CAPITULO 1

1. MANUAL TECNICO

1.1 Instalación de Software.

1.1.1 Instalación del Java Development Kit 1.5 (JDK) y (JRE).

1º- Dar click en el icono de instalación de JDK (ver figura 1).

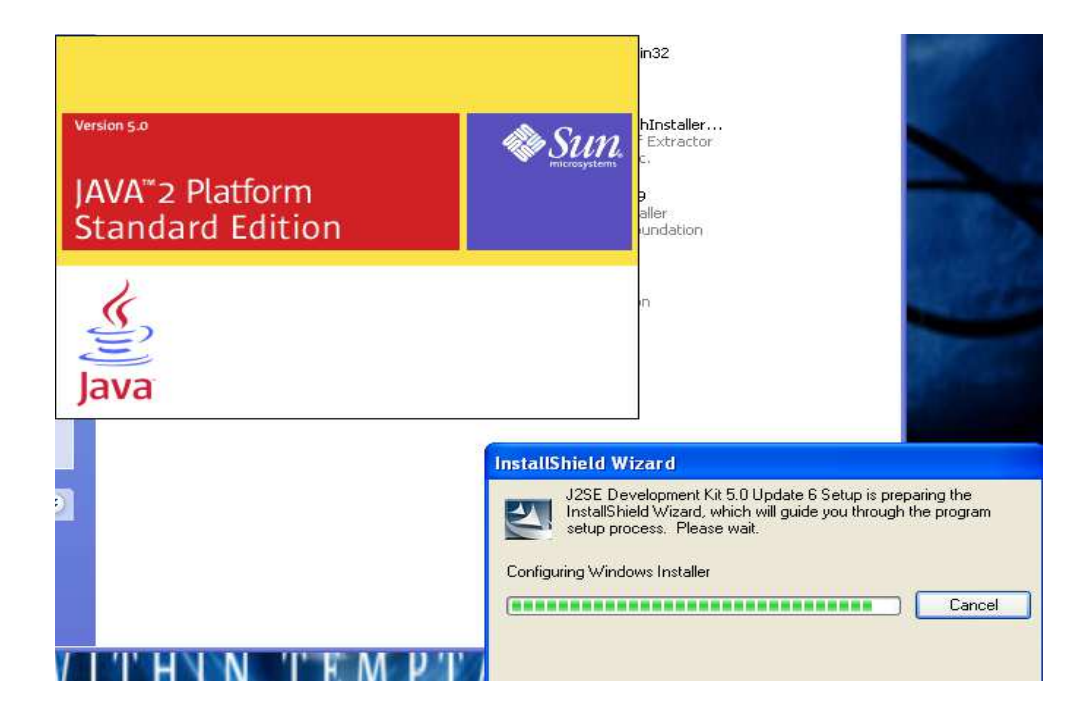


Figura 1

2º- Una vez terminada la instalación dar click finish (Ver figura 2).

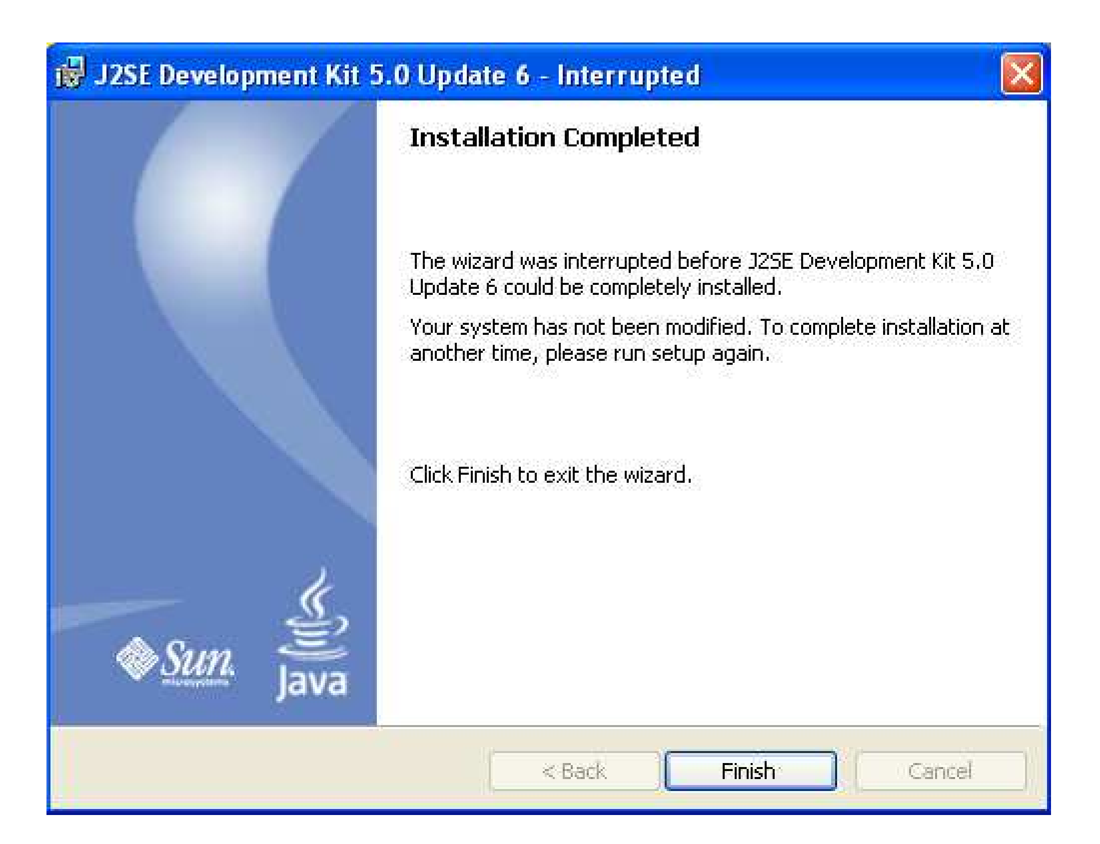


Figura 2

1.1.2 Instalación de Eclipse 3.1.

Como ya hemos comentado anteriormente, vamos a usar Eclipse como entorno de desarrollo.

Para descargar Eclipse, podemos acceder a la siguiente dirección:

http://www.eclipse.org/downloads/

De aquí podemos descargar la última versión estable del Eclipse SDK.

Para instalar Eclipse, basta descomprimir el archivo zip correspondiente en la ruta deseada. Se creará la estructura de directorios necesaria para poder lanzar Eclipse (Ver figura 3):

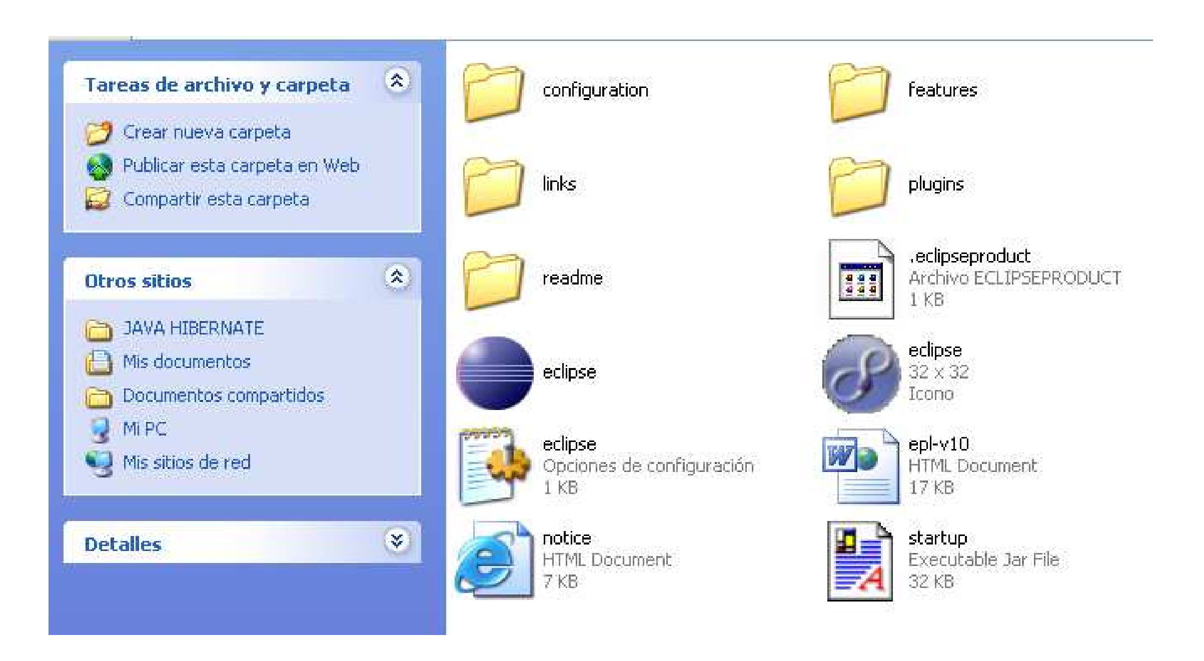


Figura 3

Para iniciar Eclipse, basta con hacer doble click sobre el archivo eclipse.exe.

1.1.3 Instalación de Servidor Web Apache Tomcat 5.5.

Dar doble clic sobre el icono de instalación del Tomcat

1º- Primera ventana. Pulsa el boton Next (Ver figura 4).



Figura 4

2º- Después de leer y estar de acuerdo con los términos de uso dar click en el boton I Agree (Ver figura 5).

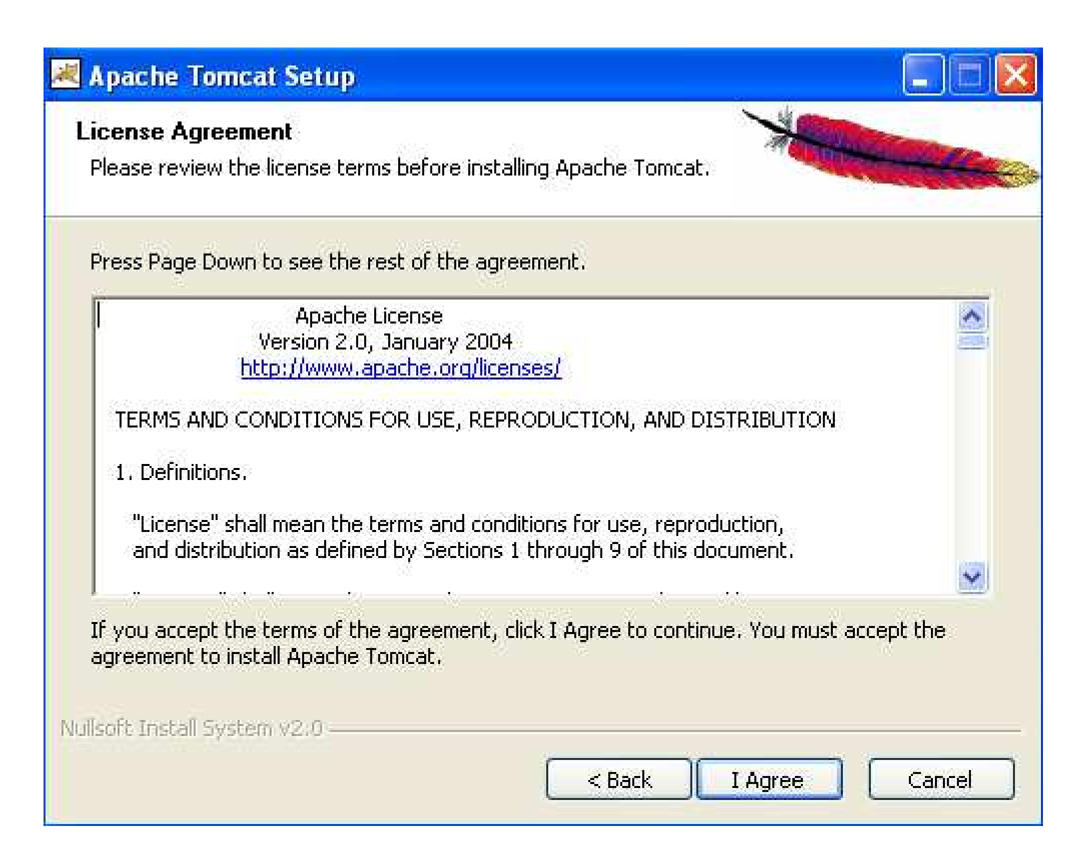


Figura 5

3º. Elegir los componentes que desea instalar, por default use la selección predeterminada (ver figura 6).

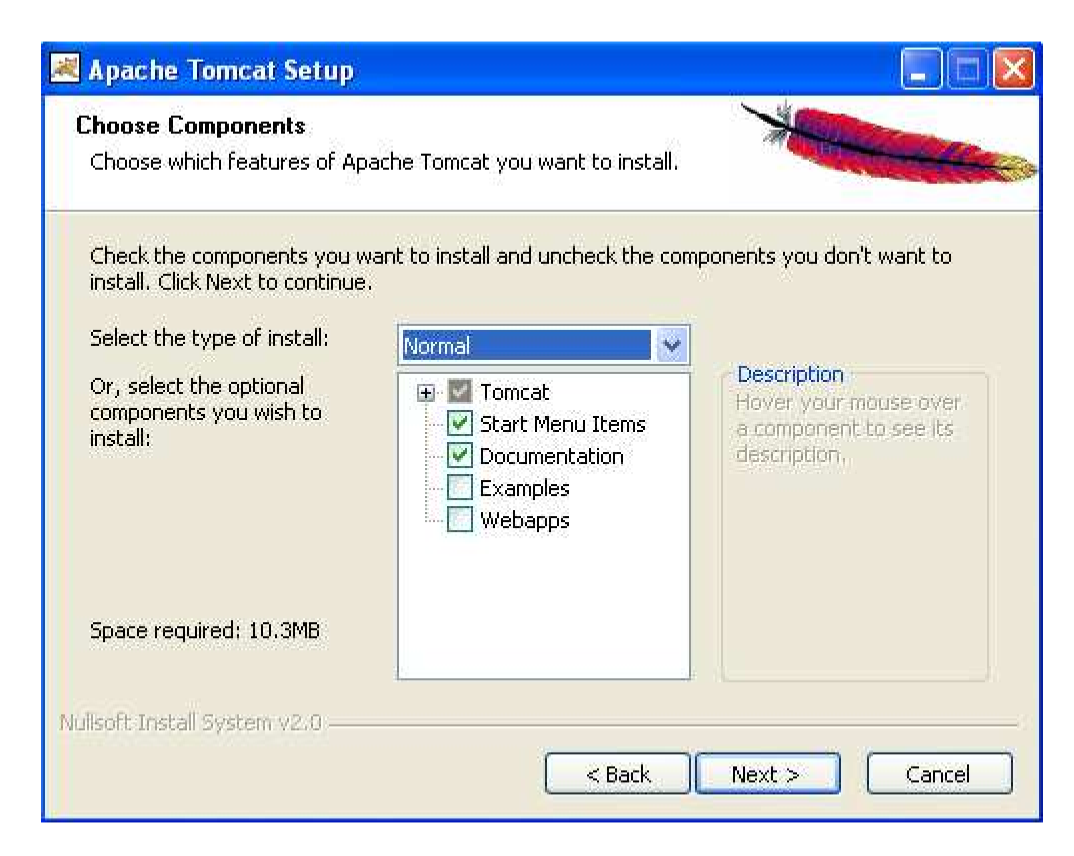


Figura 6

4º- Usted puede elegir la ruta en donde se instalará el tomcat, pero la que el instalador define es la mas adecuada (Ver figura 7).

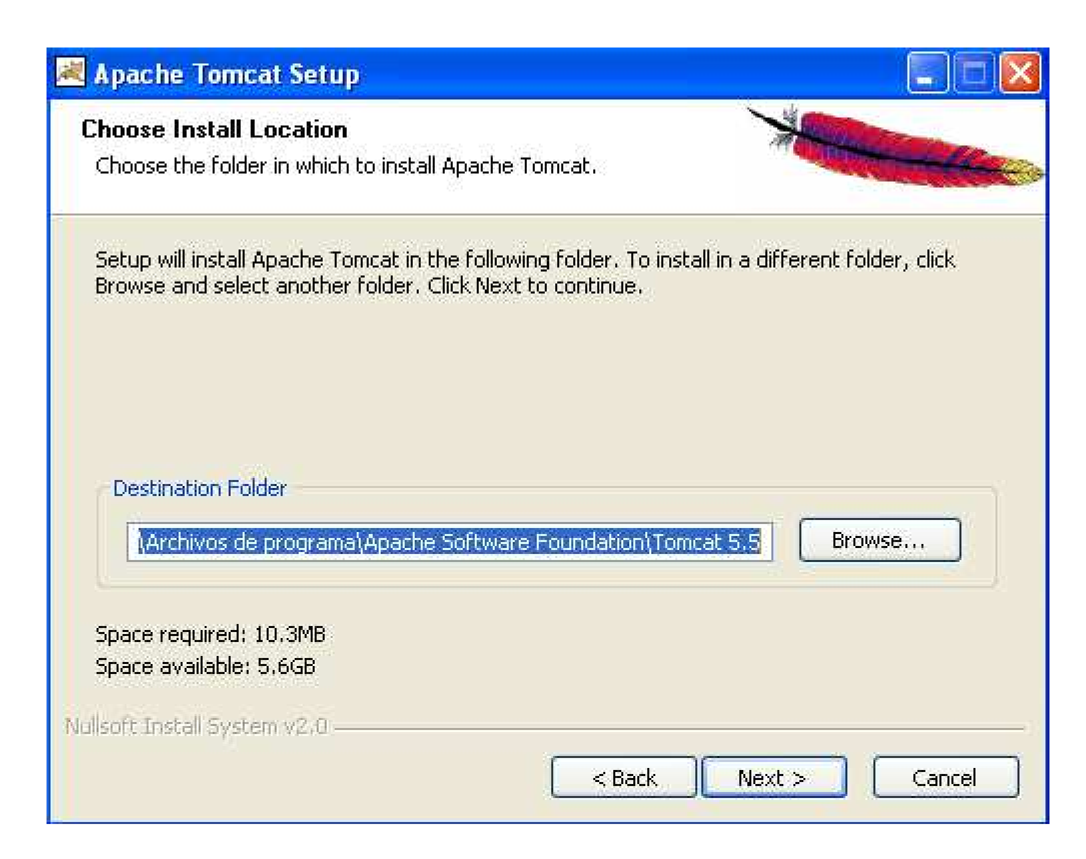


Figura 7

5º- Una vez elegida la ruta debemos configurar el puerto que utilizar y definir un usuario y una contraseña para el acceso (Ver figura 8).

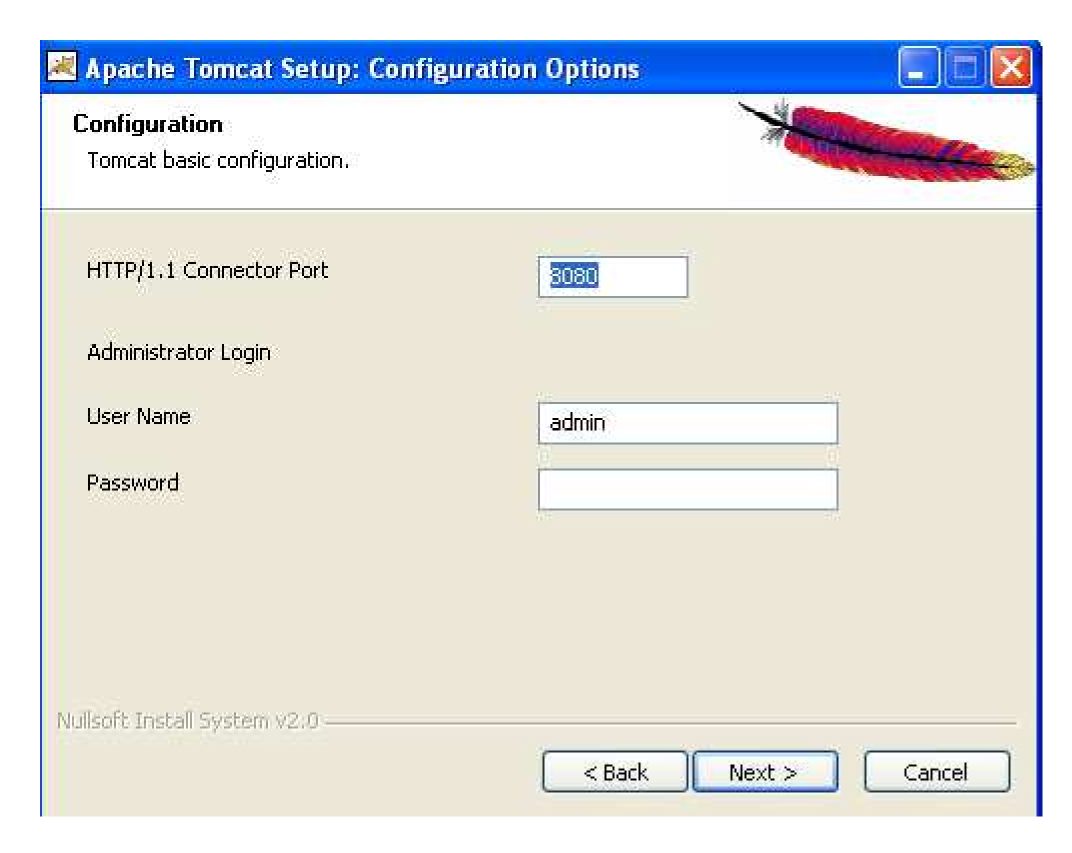


Figura 8

6º- Se debe elegir la ruta en donde se encuentre instalado el JRE de java, y luego dar click en install para que proceda con la instalación (Ver figura 9).

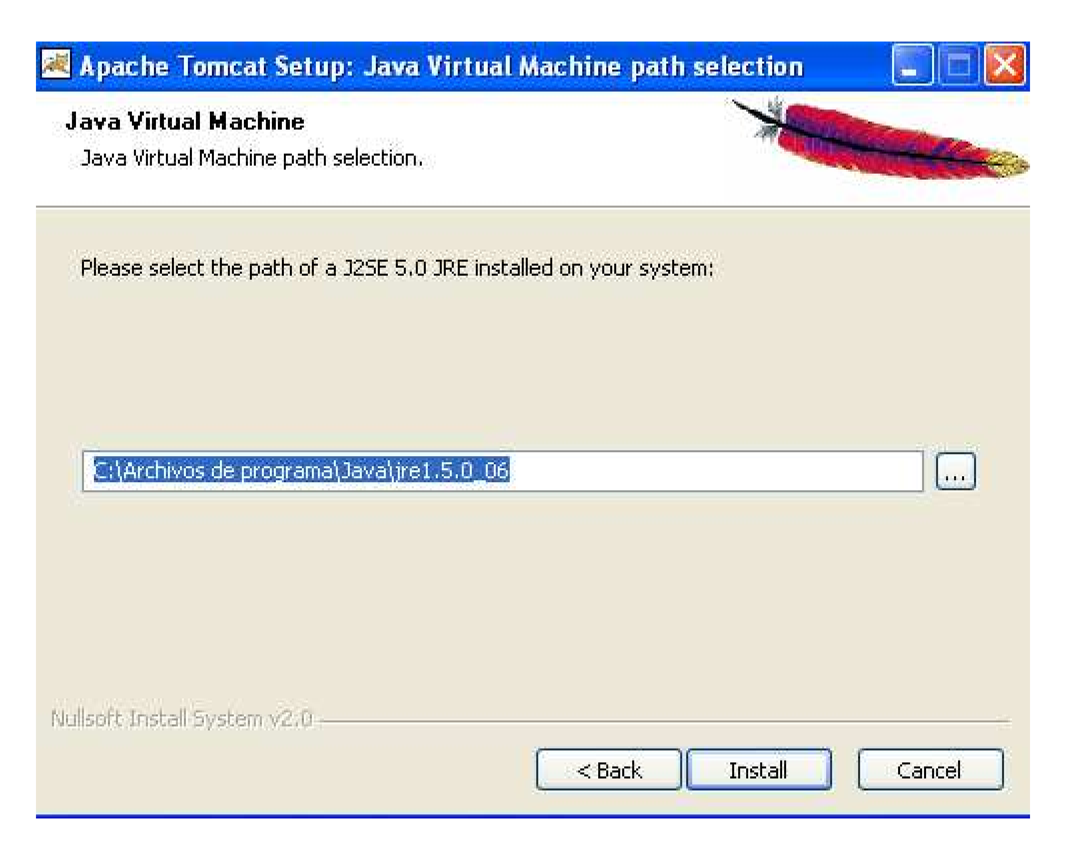


Figura 9

7º- Esperar a que termine de copiar todos los archivos (Ver figura 10).

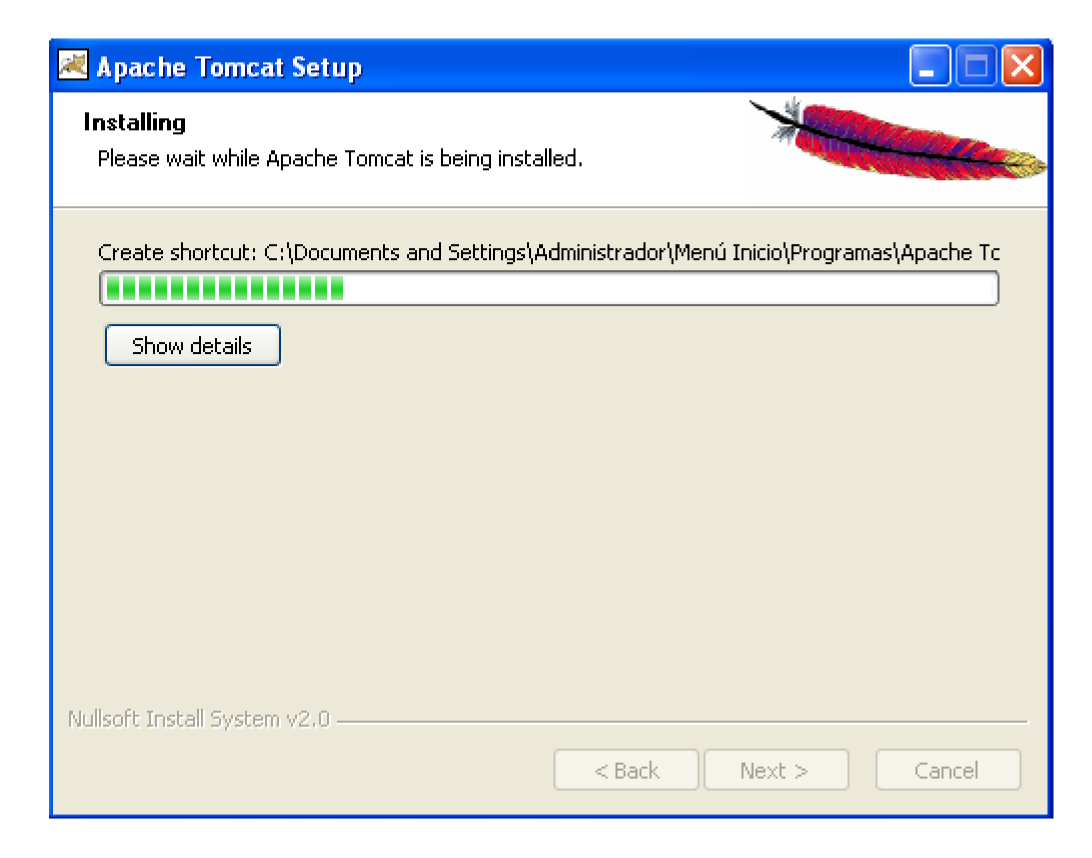


Figura 10

8º- Luego dar click en finish para finalizar con la instalación (Ver figura 11).



Figura 11

1.1.4 Instalación de Servidor de Correo Argosoft 1.8.

En menos de 10 minutos y de forma muy sencilla vamos a instalar un servidor de Correo.

Es un archivo autoextraible, por lo que haz dos veces click sobre el fichero que te has bajado (se llama agsmail.exe) y sigue estos pasos.

1º- Primera ventana. Pulsa el botón SETUP (Ver figura 12).

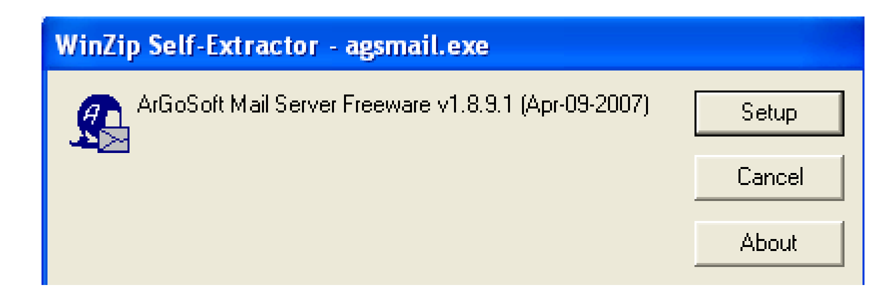


Figura 12

2º Pantalla de bienvenida del instalador del programa. Pulsamos Next (Ver figura 13).

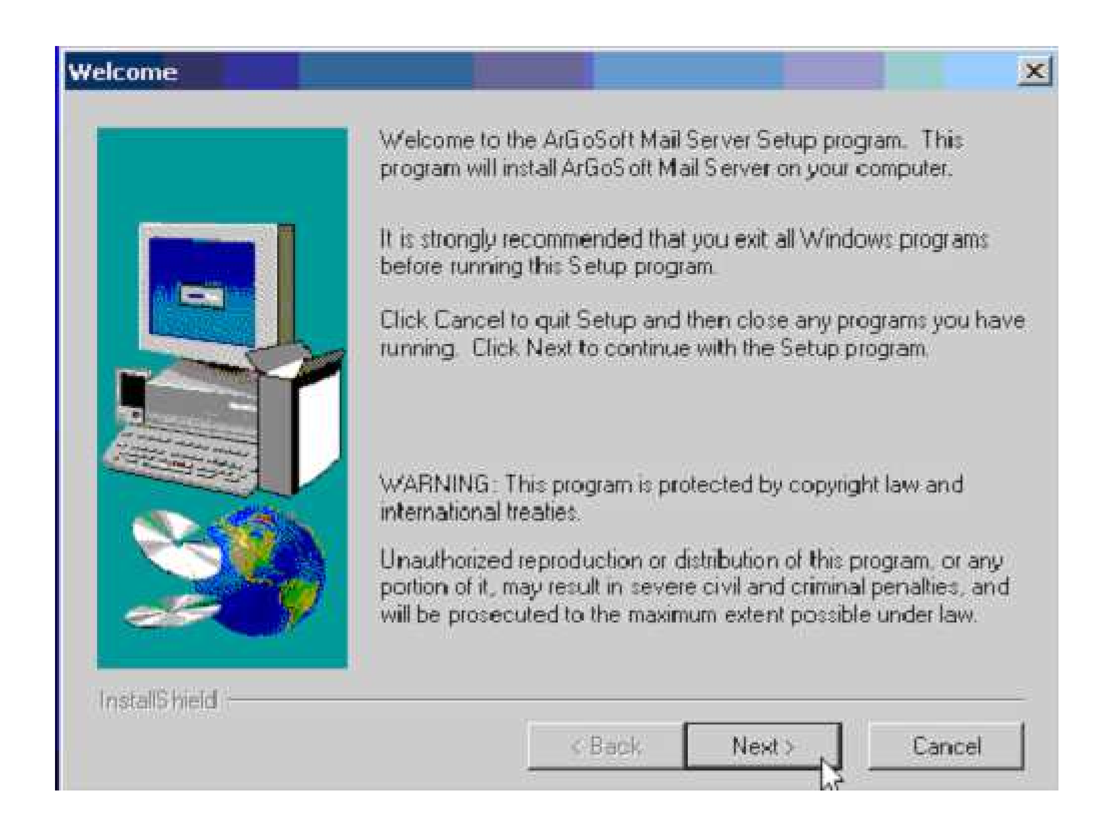


Figura 13

3º A continuación viene la decisión del directorio donde vamos a instalarlo. A mi me parece bien el que nos propone, pero si quieres cambiarlo, solo tienes que escribir el directorio donde se instalará el programa. Pulsamos sobre el botón Next (Ver figura 14).



Figura 14

4º Ahora nos pregunta donde queremos poner los iconos de acceso al programa en el Menu de Inicios, programas. Pulsamos el botón Next (Ver figura 15).



Figura 15

5º Ya ha tomado todos los datos que necesita y nos lo muestra para ver si son correctos. Pulsa Next (Ver figura 16).

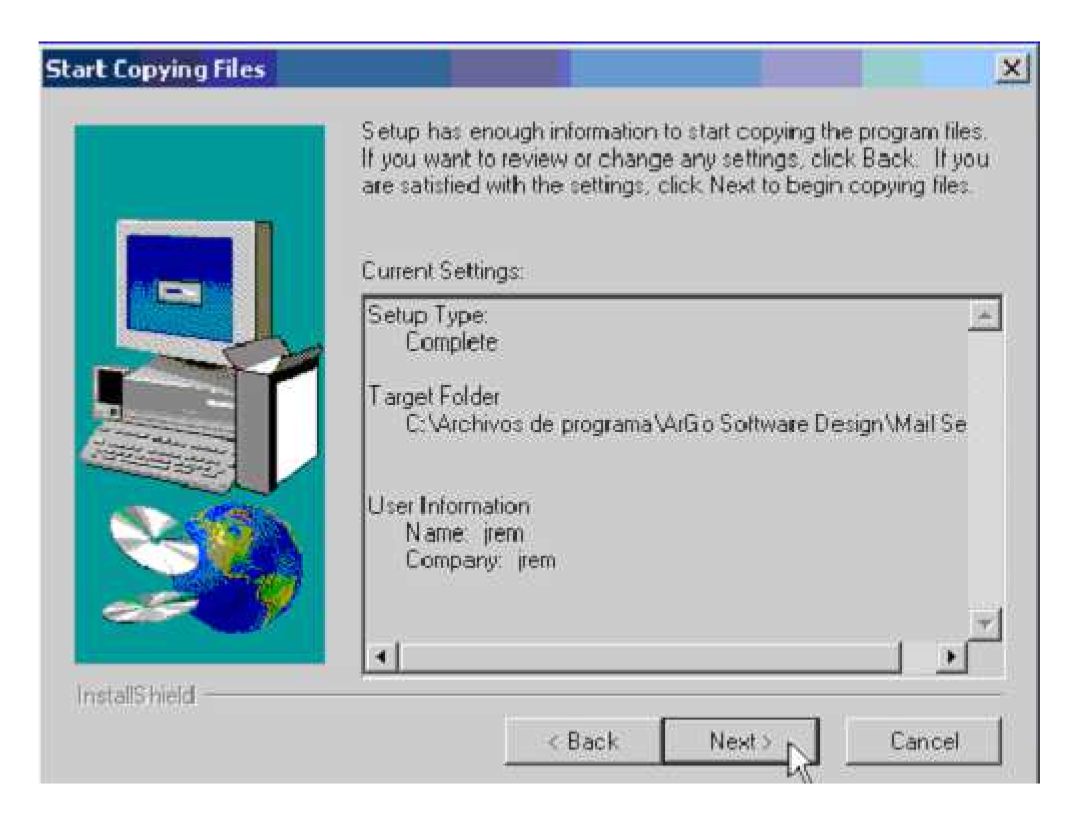


Figura 16

6º Ahora, ya por fín, nos dice que ha terminado y que para arrancar el programa lo hagamos desde donde hemos instalado los iconos. Pulsar Finish (Ver figura 17).

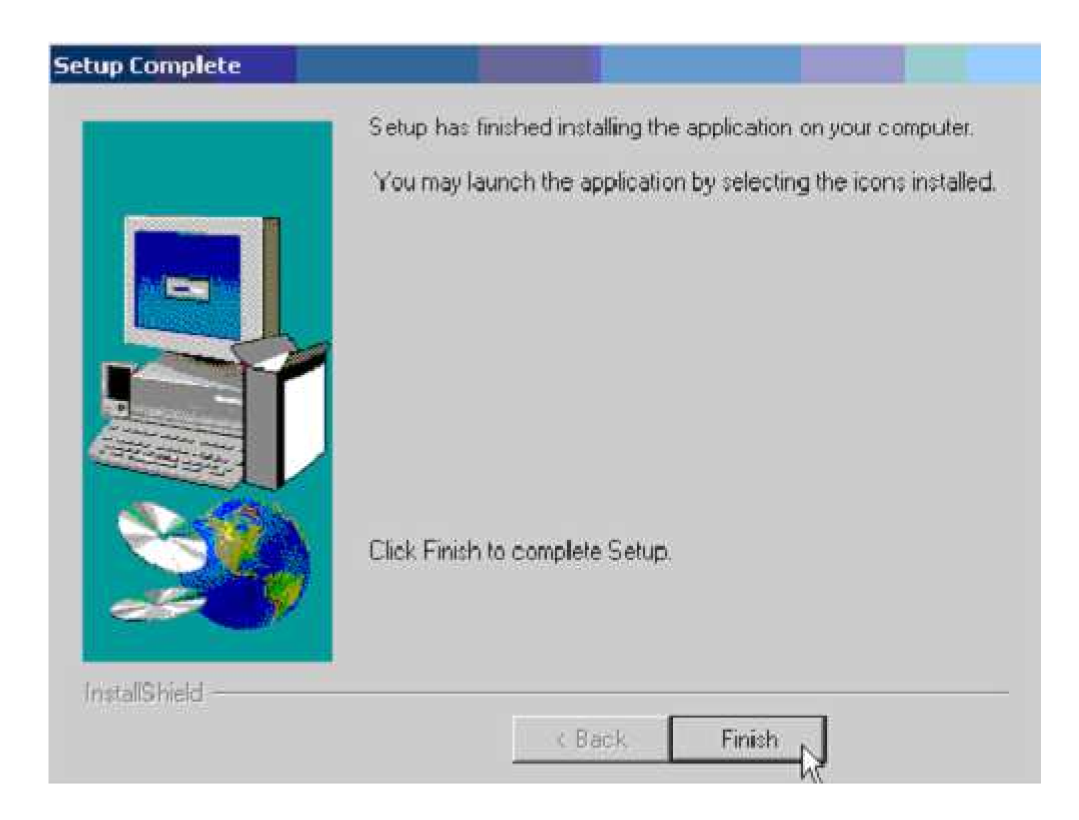


Figura 17

7º Vamos, pues a arrancar el programa. Vamos al Menu de Inicio, Programas, Argosoft Mail Server, pulsamos sobre ArgoSoft Mail Server(Ver figura 18).



Figura 18

8º Nos hace la advertencia de que pongamos el DNS del servidor. Pulsar OK (Ver figura 19).

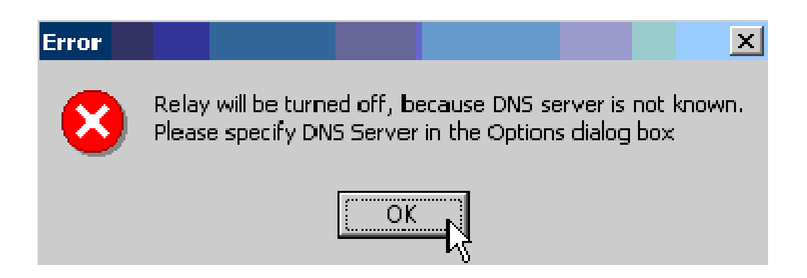


Figura 19

9º Entramos en el programa y vamos a configurarlo. Primero vamos a Tools, Options (Ver figura 20).

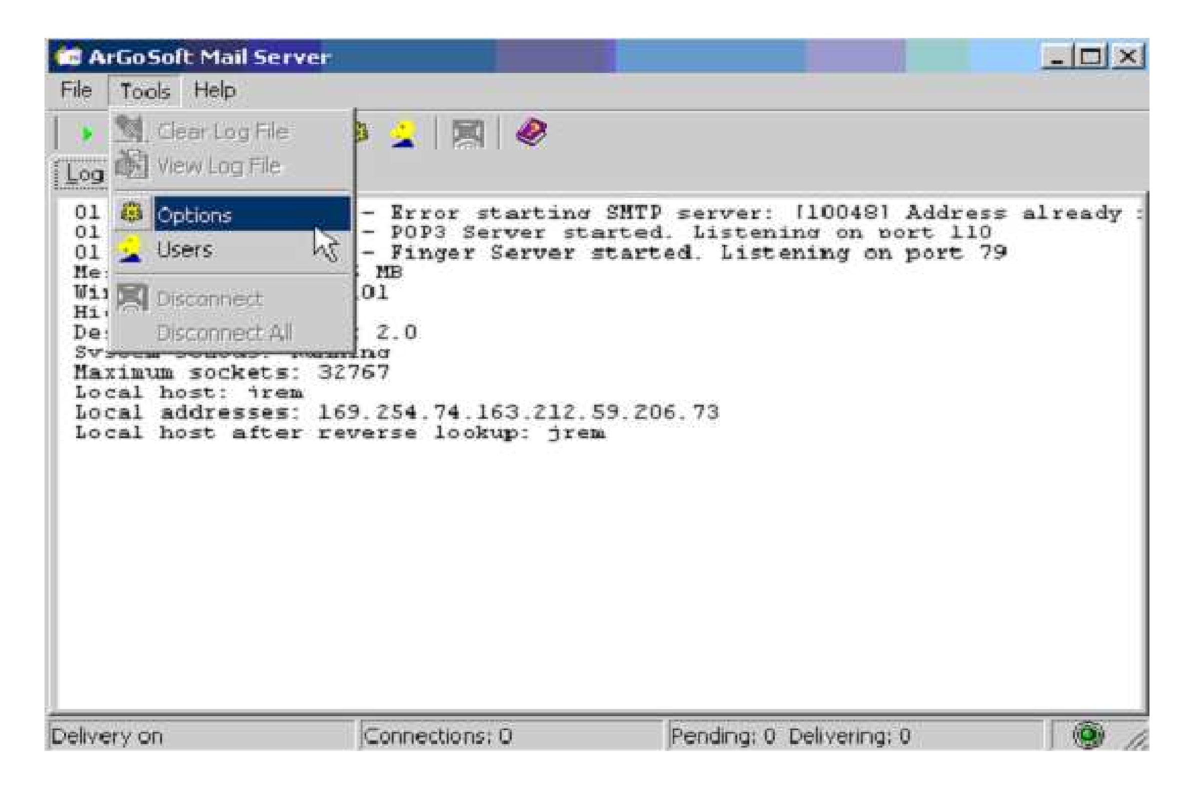


Figura 20

10º Vamos a las opciones del servidor. Solo nos hace falta la ficha de General (aunque las otras tienen su utilidad no vamos a entrar en ellas). Bien, en el campo DNS Server poner la dirección IP de nuestro servidor de dominio.

Pulsar la opcion de (Allow Relay) La opción de Automatically Start the server sirve para poner en marcha el servidor nada mas arrancar nuestro Windows. Como medida de precaución te aconsejo que no lo hagas, y que pongas en marcha el servidor solo cuando vayas a enviar un correo. La razon es la propagación de virus. Si un virus se interna y quiere reproducirse, no podrá reenviarse a ningun lugar si el servidor está detenido. Pulsar OK (Ver figura 21).

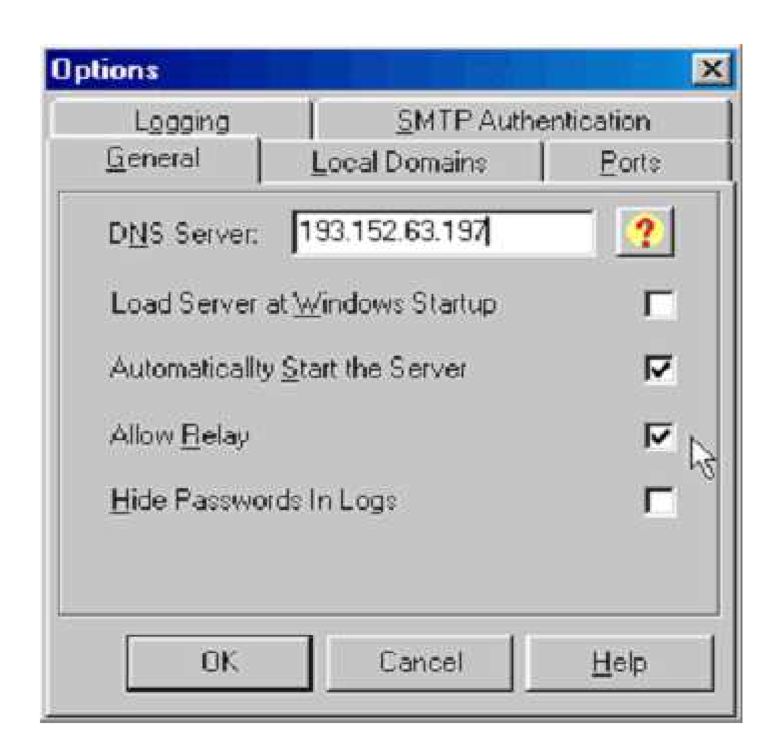


Figura 21

11º Para poner en marcha el servidor o pararlo, la primera vez, lo tienes que hacer desde el icono mostrado a continuación (el de parado es un cuadrado rojo). Cada vez que pongáis en marcha el PC, el servidor se pondra en funcionamiento (Ver figura 22).



Figura 22

12º Bien, el servidor está corriendo.Puedes ver que en la parte inferior derecha hay un nuevo icono. Nos dice que está funcionando (Ver figura 23).



Figura 23

Ahora tenemos que configurar nuestro programa de correo. Para ello tienes que ir a las propiedades de la cuenta y cambiar el campo SMTP por 27.0.0.1 o poniendo localhost (es lo mismo) En el Outlook Express, se llega desde el Menú Herramientas, Cuentas. Se eligen todas las cuentas que tengas de correo una por una (o las que quieras utilizar) y pulsas sobre Propiedades.

En el campo Correo Saliente pones 127.0.0.1, Una vez hecho, Aplicar y Aceptar. Ya esta, ya tienes funcionando un servidor propio de correo saliente.

Para los demás clientes de correo la mecánica es la misma.

1.1.5 INSTALACION DE BASE DE DATOS ORACLE 9I

Introduciremos el CD 1 de la instalación, los tres CDs que componen el programa de instalación se pueden descargar gratuitamente (siempre que no sea con fines lucrativos) desde la propia web de oracle: www.oracle.com (ver figura 24).



Figura 24

• Tras seleccionar la ruta de los ficheros de oracle pulsaremos en *Siguiente*.

• A continuación seleccionaremos el tipo de instalación que deseemos (Ver figura 25):



Figura 25

En nuestro caso, puesto que crearemos la base de datos en el servidor donde estamos realizando la instalación seleccionaremos *Oracle9i Database 9.2.0.1.0* . Si ya disponemos de un servidor de Oracle con las correspondientes bases de datos a las que queramos acceder será suficiente con seleccionar la opción *Oracle9i Client 9.2.0.1.0* , en este caso la instalación es bastante más simple, pues sólo es necesario especificar la IP o

el nombre de red del servidor de Oracle así como el nombre (sid) de la base de datos a la que queramos acceder.

• Seleccionaremos el tipo de instalación que queramos realizar (Ver figura 26):

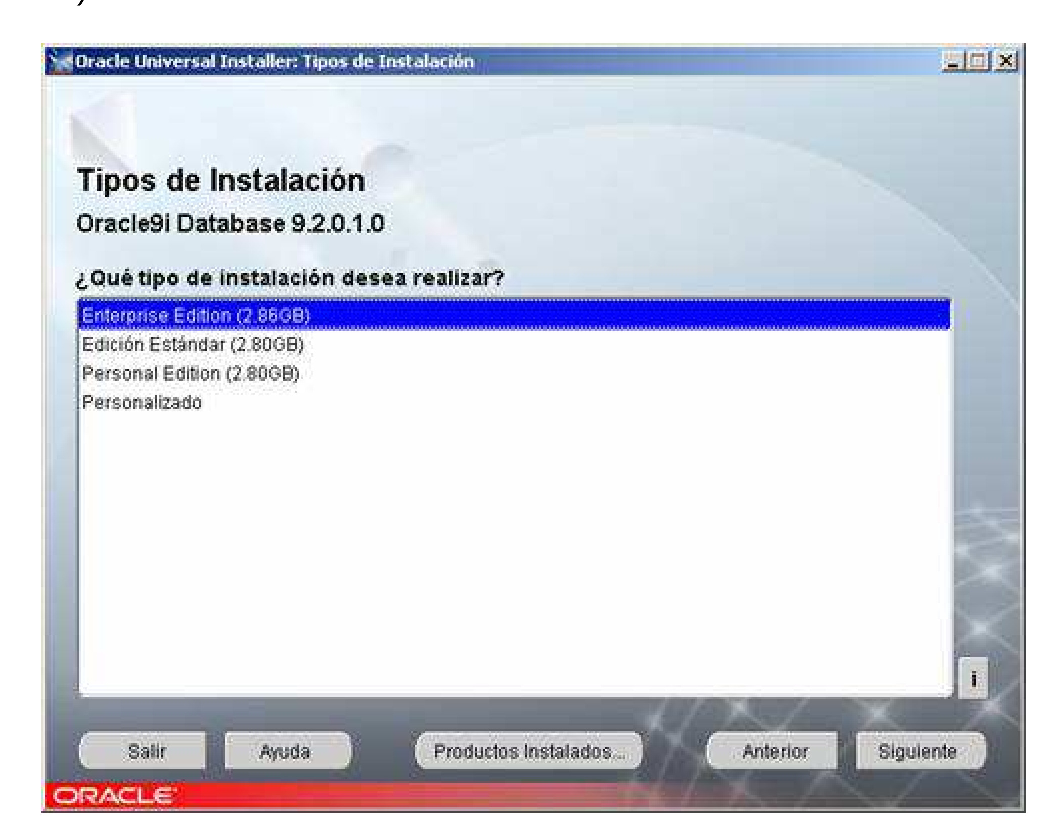


Figura 26

En nuestro caso, seleccionaremos *Enterprise Edition* , si queremos realizar una instalación más avanzada (especificando manualmente las opciones a instalar) seleccionaremos *Personalizado* , en este caso aparecería una ventana como esta (Ver figura 27):

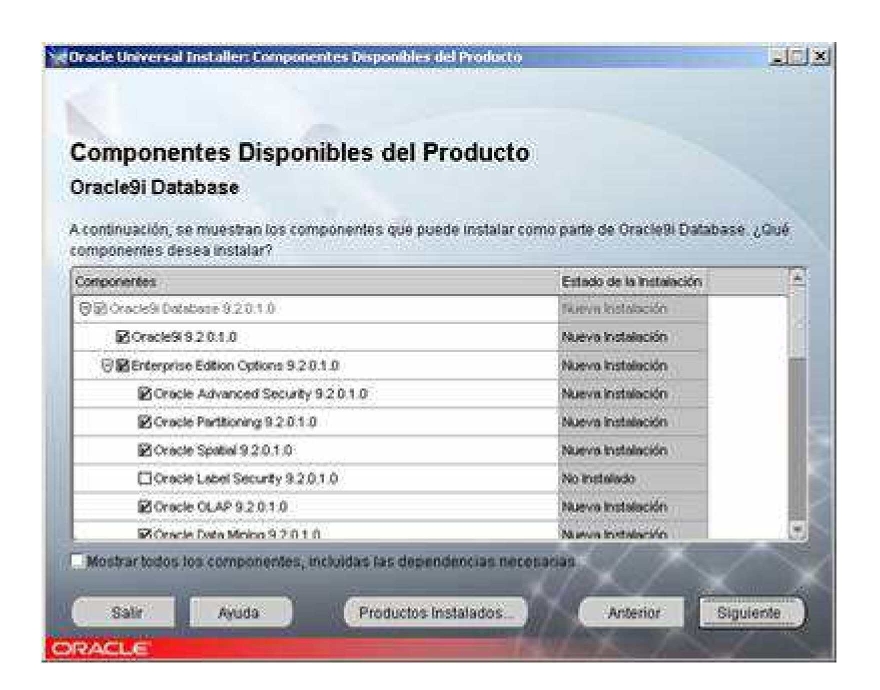


Figura 27

• Seleccionaremos la siguiente opción dependiendo del uso que le queramos dar a nuestra base de datos, normalmente es para *Uso General* . Si no queremos crear una base de datos en el proceso de instalación (se puede crear en otro momento) seleccionaremos *Sólo software (Ver figura 28)*:



Figura 28

Seleccionaremos el puerto para Oracle MTS Service, normalmente se suele seleccionar el puerto por defecto *2030* . Este parámetro es muy importante pues, si decidimos cambiar el puerto por defecto, cuando queramos que un cliente se conecte al servidor deberemos especificar el puerto que hayamos seleccionado en este punto de la instalación (Ver figura 29):



Figura 29

• Especificaremos el nombre de la base de datos (con un máximo de 8 caracteres), (Ver figura 30):



Figura 30

Nota: el *SID* es el identificador interno que utilizará Oracle para referenciar a nuestra base de datos, se puede elegir uno diferente al del nombre de la Base de Datos, aunque se suele utilizar el mismo.

• En este punto de la instalación seleccionaremos la ubicación de los archivos de la base de datos que la instalación creará. Oracle recomienda que la ubicación de los archivos de la base de datos esté en un disco físico distinto al de los archivos de la instalación (software de Oracle). También recomienda que los archivos de Redo Log estén multiplexados (varias

copias, esto se configura en la consola de administración de Oracle) y en diferentes discos físicos. Lógicamente es lo recomendable por Oracle y sólo se configura así cuando se trata de una Base de Datos que tendrá múltiples accesos concurrentes (al mismo tiempo) y con un volumen de datos importante, pues el desembolso económico en harware para la correcta instalación de Oracle puede ser importante. A pesar de todo no hay ningún problema por instalar la base de datos en el mismo disco duro que el software de Oracle. Si nuestra organización dispone, por ejemplo, de unos 20 usuarios conectados a Oracle no habría una pérdida del rendimiento por instalarla en el mismo disco duro. Por supuesto esto es orientativo pues dependerá también del volumen de datos que necesite cada usuario así como de otros factores (rpm del disco duro, características del servidor de Oracle (procesadores, memoria RAM, …), velocidad de la red local, tipo de conexión que realice el software de nuestra empresa que acceda a Oracle (el acceso nativo sin necesidad de utilizar controladores ODBC incrementa considerablemente el rendimiento frente a accesos mediante controladores ODBC que no dejan de ser una pasarela entre el software y Oracle):



Figura 31

• Seleccionaremos el juego de caracteres que vayamos a utilizar, si dejamos el *juego de caracteres por defecto* Oracle utilizará el juego de caracteres que tengamos configurado en nuestro sistema operativo, es la opción recomendada. Este parámetro sólo sería importante en el caso en que queramos exportar nuestra base de datos a otro servidor de Oracle, en este caso será importante que el juego de caracteres del servidor que recibirá los datos coincida con el que los exportó. Si no coinciden podrían aparecer erróneamente caracteres como "€" alojados en nuestra base de datos (ver figura 32):

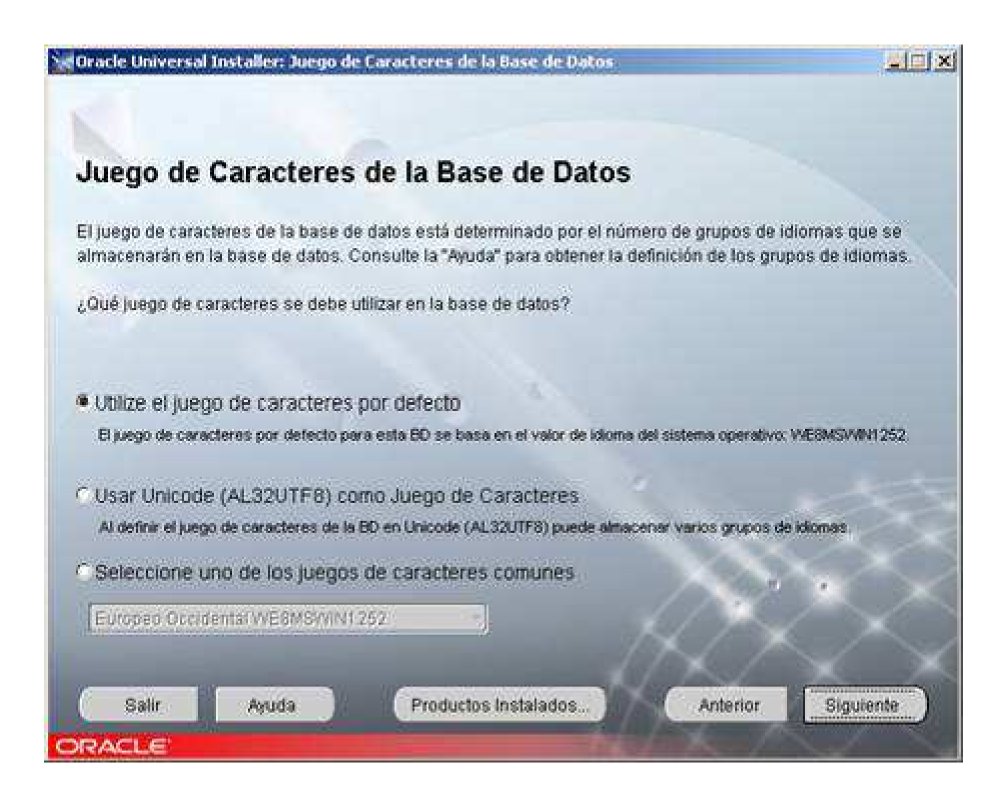


Figura 32

• Como último paso de la preinstalación nos aparecerá una ventana con el software que se va a instalar, tras comprobar que es correcto pulsaremos en *Instalar (Ver figura 33)* :



Figura 33

• Introduciremos los CDs de instalación conforme los vaya pidiendo el programa (Ver figura 34):



Figura 34

• Tras la instalación de Oracle y la creación de la base de datos aparecerá una ventana indicando que el proceso de instalación ha finalizado (Ver figura 35):



Figura 35

En esta ventana nos indica que ha finalizado el proceso de instalación y que podemos acceder desde cualquier otro PC de la red introduciendo en el navegador de Internet:

http://nombre_o_IP_del_servidor_oracle:7778

El acceso vía Internet está permitido pues al seleccionar la instalación *Enterprise Edition* se instala e inicia por defecto *Oracle http Server* (servidor web de Oracle).

Si al intentar acceder a la *Consola de Oracle Enterprise Manager* nos aparece este error (Ver figura 36):



Figura 36

Deberemos configurar el Listener manualmente, para ello accederemos a *Inicio – Programas – Oracle – OraHome92 – Configuration and Migration Tools – Net Configuration Assistant (Ver figura 37 y 38)* :

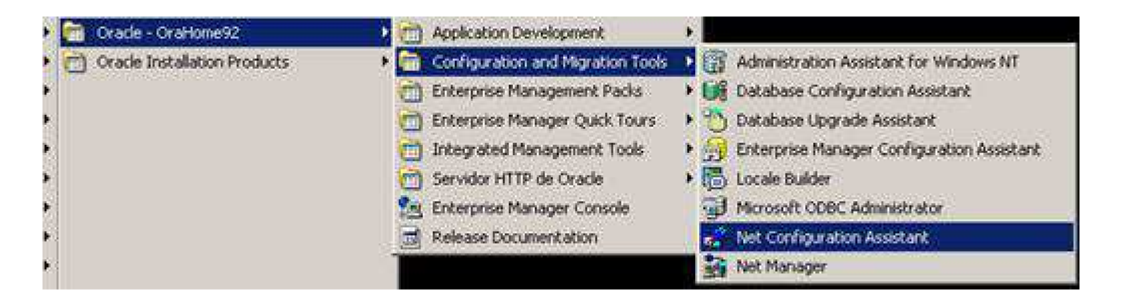


Figura 37



Figura 38

Seleccionamos *Configuración de Listener* y a continuación *Volver a Configurar (Ver figura 39):*

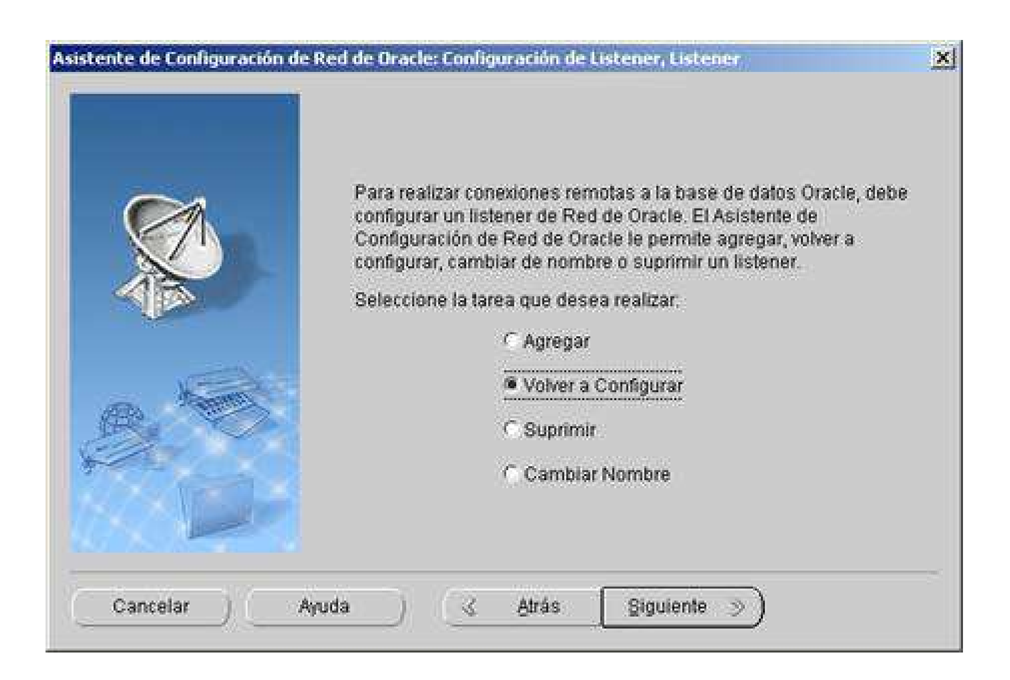


Figura 39

Nos aparecerá el Listener que el programa de instalación de Oracle crea automáticamente (ver figura 40):



Figura 40

Seleccionamos el protocolo a utilizar (TCP) (Ver figura 41):

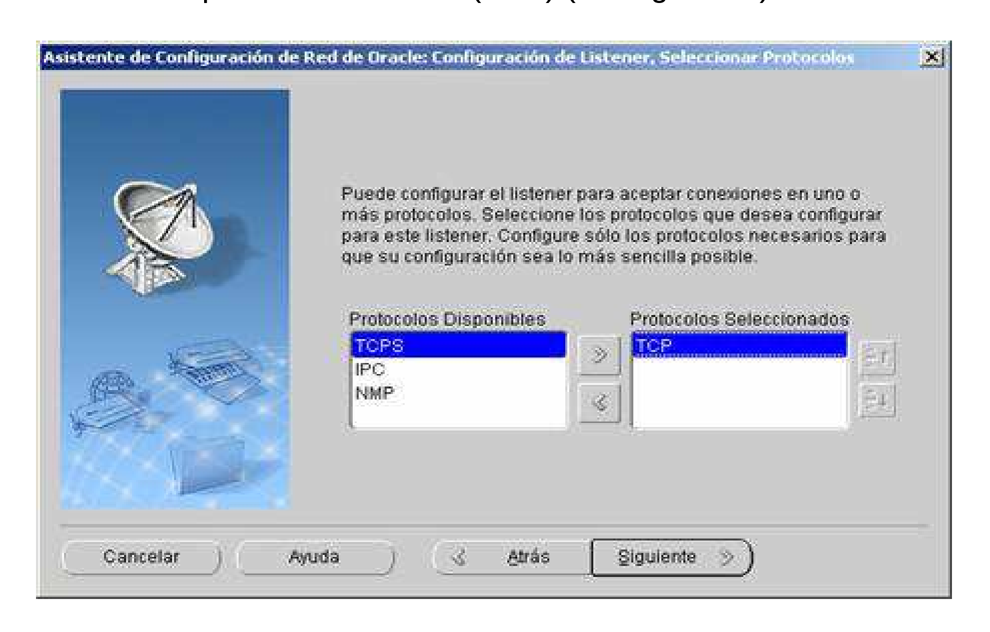


Figura 41

Seleccionamos el puerto a utilizar, por defecto 1521 (Ver figura 42):

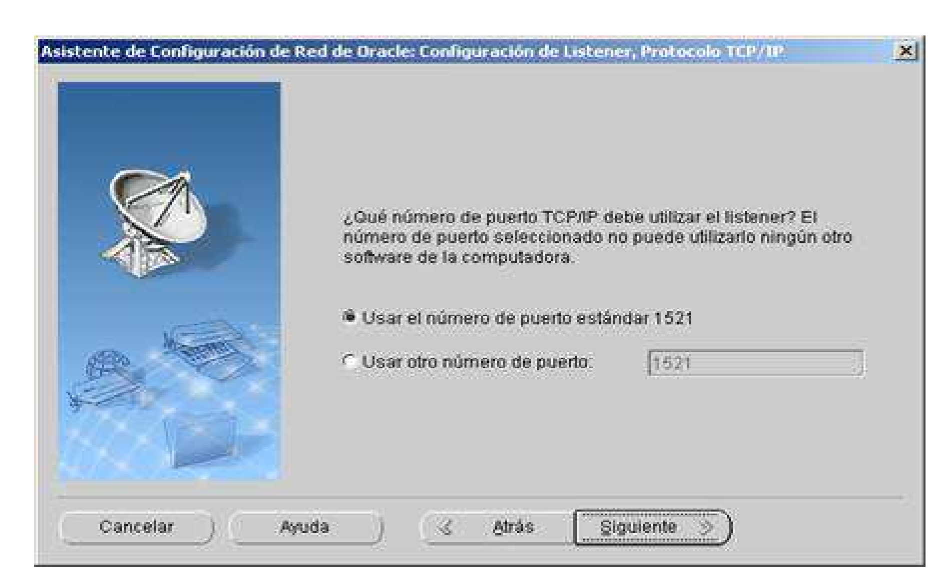


Figura 42

Con esto habremos vuelto a actualizar el Listener (Ver figura 43):



Figura 43

ALGUNOS ERRORES AL INICIAR ORACLE Y SU SOLUCIÓN

• Al iniciar la *Enterprise Manager Console* o al intentar conectarme a la Base de Datos de Oracle aparece el siguiente error (Ver figura 44 y 45):

TNSLSNR.EXE ha detectado un problema y debe cerrarse.

Más información: szAppName: TNSLSNR.EXE szAppVer: 0.0.0.0 szModName: orantcp9.dll



Figura 44

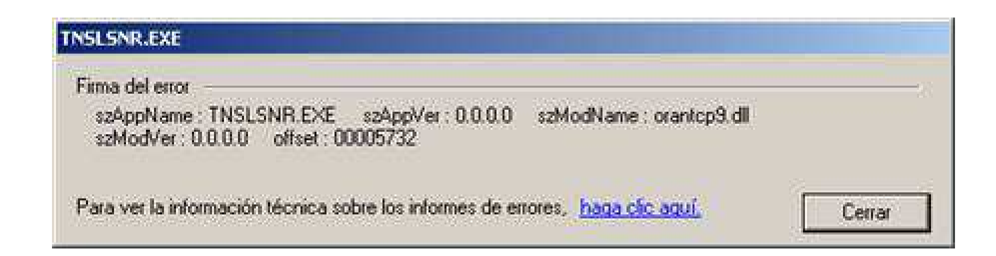


Figura 45

Y tras pulsar en *Cerrar* el servicio *OracleOraHome92TNSListener* se detiene.

Solución:

Accedemos al Regedit, desde el botón *Inicio – Ejecutar* ponemos *regedit* y pulsamos en *Aceptar* : buscamos la siguiente clave:

HKEY_LOCAL_MACHINE/SOFTWARE/ORACLE/HOME0

E insertamos una nueva entrada de tipo *Valor de cadena expandible* con el valor *TRUE(Ver figura 46)*:

USE_SHARED_SOCKET=TRUE

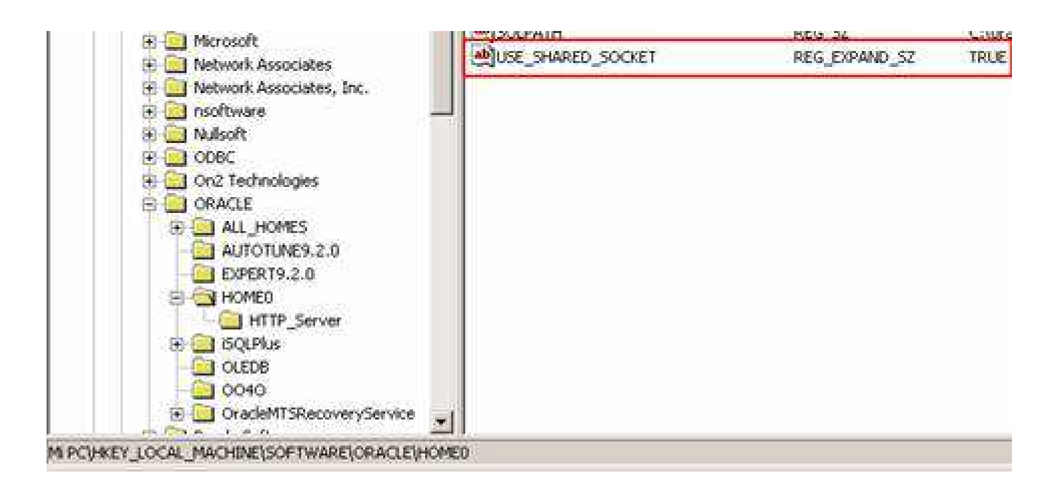


Figura 46

Una vez añadida esta clave reiniciaremos el sistema. En teoría ya debería haberse solucionado el problema.

Si el problema sigue produciéndose tras reiniciar, será conveniente comprobar el fichero ***WinsockXPFix.exe***.

• El espacio en disco no para de decrecer, hasta quedarse sin espacio. Este problema es producido por el *Servidor HTTP de Oracle Basado en Apache* , cuando dicho servidor encuentra algún problema con el Listener (como el descrito en el punto 1), lo informa escribiendo en el log que crea a tal efecto. Dicho fichero de log va creciendo desproporcionadamente hasta dejar el disco duro sin espacio. La línea del fichero de log podría ser como esta:

[warn] FastCGI: server "c:/oracle/ora92/bin/isqlplus" can't be restarted:exceeded max restarts

Solución:

Para solucionar este problema, en primer lugar deberemos detener el servicio *OracleOraHome92HTTPServer* y deshabilitarlo para que no se vuelva a iniciar, al menos, hasta que solucionemos el problema, a continuación eliminaremos el fichero de log de Apache ubicado en:

C:/oracle/ora92/Apache/Apache/logs

Eliminaremos el fichero *error_log* , si no nos deja eliminarlo será porque el servicio no se ha detenido correctamente, con lo cual deberíamos deshabilitar dicho servicio y reiniciar el equipo.

Con esto hemos solucionado el problema parcialmente pues, aunque ya no vamos a tener problemas con el espacio en disco, el servidor de Apache estará deshabilitado, con lo cual no podremos acceder a Oracle vía Internet.

Para poder iniciar el servicio de Apache sin que deje el disco sin espacio editaremos el fichero:

C:/oracle/ora92/Apache/Apache/conf/httpd.conf

Y modificaremos la línea ***LogLevel*** *warn* por ***LogLevel error***. De esta forma evitaremos que el log se llene de *warning* y sólo se guardarán los errores. Una vez guardado el fichero *httpd.conf* podremos volver a iniciar el servicio *OracleOraHome92HTTPServer.*

• Al intentar conectar a Oracle aparece el error "TNS-12535: TNS:timeout de la operación" ó "ORA-12535: TNS:operation timed out". Dicho error se puede comprobar en el directorio de instalación de Oracle:

C:/oracle/ora92/bin/sqlnet.log

Solución:

Normalmente es debido a un fallo de conexión de red con el servidor. Si, tras comprobar que la red funciona adecuadamenete, el problema sigue apareciendo se puede deber también a la intervención de un cortafuegos (tanto hardware como software). Por ejemplo, si tenemos instalado Oracle Server en un PC con Windows XP, el cortafuegos software que éste incorpora inpedirá el acceso a la Base de Datos Oracle desde otros PCs. Para solucionarlo deberemos configurar adecuadamente en cortafuegos abriendo los puertos que utiliza Oracle, tales como el 1521.

INICIAR / PARAR / PROBAR LA BASE DE DATOS MANUALMENTE

Para iniciar una base de datos de Oracle manualmente seguiremos los siguientes pasos:
Desde la línea de comandos (consola de MS-DOS) pondremos:

Sqlplus /nolog

(para conectarnos a SQLPlus que es la consola de Oracle para MS-DOS, útil para realizar tareas de mantenimiento de la Base de Datos, tales como reparación, apagado manual, inicio manual, …).

Connect / as sysdba

(para conectarnos a Oracle como administradores, al no poner nombre de usuario delante de “/” se asume el usuario del sistema operativo (éste ha de ser administrador)).

Select status from v$instance

(con esta consulta comprobamos que la base de datos no está iniciada, para ello deberá devolver un valor distinto a *STARTED*).

Select database_status from v$instance

startup nomount

(con esta sentencia iniciaremos la base de datos sin montar, puede ser útil para reparar algún *datafile* dañado o para realizar tareas que no se podrían llevar a cabo con la base de datos montada)

shutdown immediate

(cierra la base de datos, incluso aunque haya usuarios conectados)

startup

(inicia y monta la base de datos, la deja lista para ser usada)

alter user hr identified by hr account unlock

(esta instrucción sirve para desbloquear un usuario bloqueado y cambiarle la contraseña, en este caso hemos desbloqueado un usuario que Oracle crea por defecto, llamado hr y le hemos asignado la contraseña hr)

connect hr/hr

(nos conectamos con el usuario hr para hacer una Select y comprobar que funciona correctamente)

select * from employees

(hacemos una consulta de selección para que nos muestre el contenido de la tabla employees que Oracle crea automáticamente en la instalación)

1.2 Especificación de Paquetes y Clases.

1.2.1 Clase PageViewCountData.java (Paquete de.laures.cewolf.example)

```
package de.laures.cewolf.example;
import java.io.Serializable;
import java.util.Date;
import java.util.Map;

import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;
import org.jfree.data.category.CategoryDataset;
import org.jfree.data.category.DefaultCategoryDataset;
import de.laures.cewolf.DatasetProduceException;
import de.laures.cewolf.DatasetProducer;
import de.laures.cewolf.links.CategoryItemLinkGenerator;
import de.laures.cewolf.tooltips.CategoryToolTipGenerator;


public class PageViewCountData implements DatasetProducer,
CategoryToolTipGenerator, CategoryItemLinkGenerator, Serializable
{
private static final Log log = LogFactory.getLog(PageViewCountData.class);

private final String[] categories =    {"mon", "tue", "wen", "thu", "fri", "sat",
"sun"};
private final String[] seriesNames =    {"cewolfset.jsp", "tutorial.jsp",
"testpage.jsp", "performancetest.jsp"};

/** Produces some random data. */
public Object produceDataset(Map params) throws
DatasetProduceException
{
log.debug("producing data.");
DefaultCategoryDataset dataset = new DefaultCategoryDataset()
{
   protected void finalize() throws Throwable
        {
        super.finalize();
        log.debug(this +" finalized.");
        }
};
```

```
 for (int series = 0; series < seriesNames.length; series ++)
{
          int lastY = (int)(Math.random() * 1000 + 1000);
          for (int i = 0; i < categories.length; i++)
          {
          final int y = lastY + (int)(Math.random() * 200 - 100);
          lastY = y;
          dataset.addValue(y, seriesNames[series], categories[i]);
                              }
          }
    return dataset;
}

  /**  This producer's data is invalidated after 5 seconds. By this method the
    * producer can influence Cewolf's caching behaviour the way it wants to. */

       public boolean hasExpired(Map params, Date since)
          {
          log.debug(getClass().getName() + "hasExpired()");
          return (System.currentTimeMillis() - since.getTime())  > 5000;
          }

       /**  Returns a unique ID for this DatasetProducer  */

       public String getProducerId()
          {
                    return "PageViewCountData DatasetProducer";
          }

       /** Returns a link target for a special data item. */

       public String generateLink(Object data, int series, Object category)
          {
                    return seriesNames[series];
          }

       /** @see java.lang.Object#finalize() */

       protected void finalize() throws Throwable
          {
                    super.finalize();
                    log.debug(this + " finalized.");
          }
```

```
        /** @see
org.jfree.chart.tooltips.CategoryToolTipGenerator#generateToolTip(Category
Dataset, int, int) */

        public String generateToolTip(CategoryDataset arg0, int series, int arg2)
          {
                    return seriesNames[series];
          }
          }
```

1.2.2 Clase AccesoBasedatos3.java (Paquete sgpi)

```
package sgpi;

import java.sql.ResultSet;
import java.util.List;
import java.util.Set;
import sgpi.pojo.general.Alarma;
import sgpi.pojo.general.Historial;
import sgpi.pojo.general.Mensaje;
import sgpi.pojo.general.Participante;
import sgpi.pojo.general.Persona;
import sgpi.pojo.general.Proyecto;
import sgpi.pojo.general.Responsable;
import sgpi.pojo.general.Servidor;
import sgpi.pojo.general.TipoEnvio;


/** Interfaz de acceso a base de datos SGPI */

public interface AccesoBaseDatos
{

Integer Codigoproyecto, Integer CodigoTipoEnvio  String FechaFin, String
Repeticiones,Integer CodigoParticipante,
Integer CodigoOrigen,String RutArchivo, String NombreArchivo);

void guardaResponsable(Responsable r);
void guardaAlarma(Alarma l);
void guardaMensaje(Mensaje m);
void guardaServidor(Servidor v);
void guardaHistorial(Historial his);
}
```

1.2.3 Clase AppException.java (Paquete sgpi)

```
package sgpi;

import java.sql.Date;

public class AppException extends Exception {

	private String usuario;
	private String accion;
	private Date fecha;

	public AppException(String usuario, String accion, String detalle) {
		super(detalle);

		this.usuario = usuario;
		this.accion = accion;
		this.fecha = new Date(System.currentTimeMillis());
	}

	public AppException() {
		this(null,null,null);
	}

	public String getAccion() {
		return accion;
	}
	public void setAccion(String accion) {
		this.accion = accion;
	}
	public Date getFecha() {
		return fecha;
	}
	public void setFecha(Date fecha) {
		this.fecha = fecha;
	}
	public String getUsuario() {
		return usuario;
	}
	public void setUsuario(String usuario) {
		this.usuario = usuario;
	}

}
```

1.2.4 Clase AccesoDao3.java (Paquete sgpi.db)

```
package sgpi.db;

import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Timestamp;
import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.HashSet;
import java.util.List;
import java.util.Set;
import org.hibernate.Session;
import org.springframework.orm.hibernate3.support.HibernateDaoSupport;
import sgpi.AccesoBaseDatos;
import sgpi.pojo.general.Alarma;
import sgpi.pojo.general.Copiar;
import sgpi.pojo.general.Historial;
import sgpi.pojo.general.Mensaje;
import sgpi.pojo.general.Participante;
import sgpi.pojo.general.Persona;
import sgpi.pojo.general.Proyecto;
import sgpi.pojo.general.Responsable;
import sgpi.pojo.general.Servidor;
import sgpi.pojo.general.TipoEnvio;



public class AccesoDAO2 extends HibernateDaoSupport   implements
AccesoBaseDatos
{
public ResultSet executeQuery(String sql)
{
try
{
Session session =
getHibernateTemplate().getSessionFactory().openSession();
session.beginTransaction();
PreparedStatement statement = session.connection().prepareStatement(sql);
ResultSet result = statement.executeQuery();
session.getTransaction().commit();
return result;
}
catch (SQLException e)
```

```
{
        e.printStackTrace();
}
catch (Exception e2)
{
        e2.printStackTrace();
}
        return null;
}
public List getAlarma()
{
        return getHibernateTemplate().find("from Alarma g order by
g.codigo_alarma");
}
public List getResponsable()
{
        return getHibernateTemplate().find("from Responsable i order by
        i.codigo_respon_alarma");
}

public List getAlarma2()
{
        return getHibernateTemplate().find("from Alarma r where r.estado='P'
        order by r.codigo_alarma");
}
public List getResponsable2()
{
        return getHibernateTemplate().find("from Responsable s order by
s.codigo_respon_alarma");
}
public List getMensaje2()
{
        return getHibernateTemplate().find("from Mensaje h order by
h.codigo_mensaje");
}
public List getServidor()
{
        return getHibernateTemplate().find("from Servidor d order by
        d.codigo_servidor");
}
public List getPersonas()
        {
                return getHibernateTemplate().find("from Persona d order by
                d.nombre_completo");
        }
public List getTipoEnvio()
        {
```

```
            return getHibernateTemplate().find("from TipoEnvio t  order by
            t.codigo_tipo_envio");
      }
public List consultaTipoEnvio()
      {
            return getHibernateTemplate().find("from TipoEnvio c order by
      c.desc_envio");
      }
public List consultaProyecto()
      {
            return getHibernateTemplate().find("from Proyecto y  order by
            y.codigo_proyecto");
      }

/** Guarda responsable en la base de datos           */
public void guardaResponsable(Responsable r)
      {
            getHibernateTemplate().saveOrUpdate(r);
      }
public void guardaAlarma(Alarma l)
      {
            getHibernateTemplate().saveOrUpdate(l);
      }
public void guardaMensaje(Mensaje m)
      {
            getHibernateTemplate().saveOrUpdate(m);
      }
public void guardaServidor(Servidor v)
      {
            getHibernateTemplate().saveOrUpdate(v);
      }

public Set consultaParticipante(Integer codigo_proyecto)
      {
            Proyecto objproyecto;
            List resultados = getHibernateTemplate().find("from Proyecto p
            where p.codigo_proyecto= ?", codigo_proyecto);
            if (!resultados.isEmpty() )
            {
            objproyecto = (Proyecto) resultados.get(0);
            return objproyecto.getParticipantes();
            }
            return new HashSet();
      }

public Object saveAuditable(Object objetoAuditable)
      {
            getHibernateTemplate().saveOrUpdate(objetoAuditable);
```

```
                return objetoAuditable;
        }

public void removeObjeto(Object objeto)
        {
                getHibernateTemplate().delete(objeto);
        }
public Alarma getAlarma(Integer Alarma)
        {
                List alarmas = getHibernateTemplate().find("from Alarma t
                where t.codigo_alarma= ?", Alarma);
                if (!alarmas.isEmpty() )
                {
                        return  (Alarma) alarmas.get(0);
                }
                return null;
        }
public Mensaje getMensaje(Integer Mensaje)
        {
                return null;
        }
public Responsable getResponsable(Integer Responsable)
        {
                List responsables = getHibernateTemplate().find("from
                Responsable t where t.codigo_alarma= ?", Responsable);
                if (!responsables.isEmpty() )
                {
                return  (Responsable) responsables.get(0);
                }
                return null;
        }
public Servidor getServidor(Integer Servidor)
        {
                return null;
        }
public TipoEnvio getTipoEnvio(Integer Envio)
        {
                List envios = getHibernateTemplate().find("from TipoEnvio t
        where t.codigo_tipo_envio= ?", Envio);
                if (!envios.isEmpty() )
                {
                return  (TipoEnvio) envios.get(0);
                }
        return null;
                }
public Proyecto getProyecto(Integer Proyecto)
        {
```

```
                List proyectos = getHibernateTemplate().find("from Proyecto y
                where y.codigo_proyecto= ?", Proyecto);
                if (!proyectos.isEmpty() )
                        {
                                return  (Proyecto) proyectos.get(0);
                        }
                        return null;
        }
public Persona getPersona(Integer Persona)
        {
        List personas = getHibernateTemplate().find("from Persona n where
        n.codigo_persona= ?", Persona);
        if (!personas.isEmpty() )
                {
                        return  (Persona) personas.get(0);
                }
        return null;
        }
public Participante getParticipante(Integer Participante)
        {
                List participantes = getHibernateTemplate().find("from
                Participante n where n.codigo_participante= ?", Participante);
                if (!participantes.isEmpty() )
                {
                        return  (Participante) participantes.get(0);
                }
                return null;
        }
public List presentaAlarma(Integer Codigoproyecto)
        {
                List alarmas = getHibernateTemplate().find(
                 "select responsable.objalarma from Responsable responsable
                where responsable.objproyecto.codigo_proyecto= ? order by
                responsable.objalarma.codigo_alarma", Codigoproyecto);
                return alarmas;
        }
public List presentaResponsables(Integer Codigoproyecto)
        {
                List responsables = getHibernateTemplate().find(
                "select responsable from Responsable responsable where
                responsable.objproyecto.codigo_proyecto= ? order by
                responsable.objalarma.codigo_alarma", Codigoproyecto);
                return responsables;
        }
public void regAlarma   (String Mensaje, String Asunto,
                         Integer Codigoproyecto, Integer CodigoTipoEnvio,
```

```
                    String FechaFin, String Repeticiones,Integer
                    CodigoParticipante ,Integer CodigoOrigen,String
                    RutArchivo, String NombreArchivo) {

if (Mensaje=="" || Asunto=="" || CodigoTipoEnvio==0 ||
CodigoTipoEnvio<1 || CodigoTipoEnvio>2 ||
FechaFin=="" || CodigoParticipante==0 || CodigoOrigen==0)
{ System.out.println ("Faltan Datos .....");return; }
if (Repeticiones!="")
{
if (Repeticiones=="1" ||Repeticiones=="2"){}
else
{ System.out.println ("Repeticiones Debe ser 1-2 .....");return; }
}
Alarma alarma = new Alarma();
Mensaje mensaje = new Mensaje();
Responsable responsable = new Responsable();
Servidor servidor = new Servidor();
TipoEnvio tipoenvio = new TipoEnvio();
Timestamp fecha = new Timestamp(System.currentTimeMillis());
Calendar cal = Calendar.getInstance();
cal.setTime(fecha);
fecha = new Timestamp(cal.getTimeInMillis());

SimpleDateFormat d = new SimpleDateFormat("dd/MM/yyyy");
boolean er = false;
Timestamp fechaf  = null;
SimpleDateFormat d1 = new SimpleDateFormat("dd/MM/yyyy");
try
{
        String texto = d.format(fecha);
        fecha = new Timestamp(d1.parse(texto).getTime());
        String texto2 = FechaFin;
        fechaf = new Timestamp(d1.parse(texto2).getTime());
}
catch (ParseException e)
{
        er = true;
}
if (er || fechaf.before(fecha))
        { System.out.println ("Problemas Con Registro de Fechas
        ");return; }
if (fechaf.getMonth()==1 && fechaf.getDate()>28)
        { System.out.println ("Fecha provista es erronea");return; }

if (RutArchivo!= "" && NombreArchivo!="")
{
        try
```

```
{
Copiar.main(RutArchivo,NombreArchivo);
}catch(Exception r)
        {
        System.out.println ("Encera ");
        RutArchivo = "";NombreArchivo="";
        }
}
try
{
mensaje.setDescripcion_mensaje(Mensaje);
mensaje.setDescripcion_asunto(Asunto);
mensaje.setRuta_archivo(RutArchivo);

mensaje.setNombre_archivo(NombreArchivo);
guardaMensaje(mensaje);
}
catch (Exception err)
{
System.out.println ("Problemas Con Registro en Tabla
Mensaje ");return;
}
try
{
servidor.setDireccion_ip("127.0.0.1");
guardaServidor(servidor);
}
catch (Exception err)
{
System.out.println ("Problemas Con Registro en Tabla
Servidor ");return;
}
try
{
tipoenvio = getTipoEnvio(CodigoTipoEnvio);
alarma.setObjtipoenvio(tipoenvio);
alarma.setFecha_fin_act(fechaf);
alarma.setFecha_inicio_act(fecha);
alarma.setRepeticiones(Repeticiones);
alarma.setObjmensaje(mensaje);
alarma.setObjservidor(servidor);
alarma.setEstado("P");
guardaAlarma(alarma);
}
catch (Exception err)
{
System.out.println ("Problemas Con Registro en Tabla Alarma
");return;
```

```
            }
            try
            {
            Participante objparticipante
            =getParticipante(CodigoParticipante);
            if (objparticipante!= null)
            responsable.setObjpersona(objparticipante.getObjpersona());
            else return;
            if (Codigoproyecto != 0)
                    {
                    Proyecto objproyecto= getProyecto(Codigoproyecto);
                    if (objproyecto!= null)
                    responsable.setObjproyecto(objproyecto);
            }

                    responsable.setObjalarma(alarma);
                    responsable.setCodigo_origen(CodigoOrigen);
                    guardaResponsable(responsable);
            }
    catch (Exception err)
            {
            System.out.println ("Problemas Con Registro en Tabla
            Responsables ");return;
            }
    }

    public void guardaHistorial(Historial his)
    {
                    getHibernateTemplate().saveOrUpdate(his);
    }
}
```

1.2.5 Resultado.java (Paquete sgpi.estadisticas)

```
package sgpi.estadisticas;

import java.io.Serializable;
import java.util.Date;
import java.util.Iterator;
import java.util.List;
import java.util.Map;
import org.apache.commons.logging.Log;
import org.apache.commons.logging.LogFactory;
import org.jfree.data.category.CategoryDataset;
```

```
import org.jfree.data.category.DefaultCategoryDataset;
import sgpi.AccesoBaseDatos;
import sgpi.pojo.general.Alarma;
import de.laures.cewolf.DatasetProduceException;
import de.laures.cewolf.DatasetProducer;
import de.laures.cewolf.example.PageViewCountData;
import de.laures.cewolf.links.CategoryItemLinkGenerator;
import de.laures.cewolf.tooltips.CategoryToolTipGenerator;

public class Resultado implements DatasetProducer,
CategoryToolTipGenerator, CategoryItemLinkGenerator, Serializable
	{
		private static final Log log =
		LogFactory.getLog(PageViewCountData.class);
		private Integer codigo_proyecto;

		// These values would normally not be hard coded but
		produced by
		// some kind of data source like a database or a file
		private String[] categories;
		private String[] seriesNames ={"Puntaje"};
		private AccesoBaseDatos accesoBaseDatos;


	public AccesoBaseDatos getAccesoBaseDatos()
		{
			return accesoBaseDatos;
		}

	public void setAccesoBaseDatos(AccesoBaseDatos
	accesoBaseDatos)
		{
			this.accesoBaseDatos = accesoBaseDatos;
		}

	/** Produces some random data.  */
	public Object produceDataset(Map params) throws
	DatasetProduceException
		{
			int i = 0;
			log.debug("producing data.");
			DefaultCategoryDataset dataset = new
			DefaultCategoryDataset()
			{
			/**  @see java.lang.Object#finalize()  */
			protected void finalize() throws Throwable
			{
			super.finalize();
```

```
		log.debug(this +" finalized.");
		}
	};

		List lista =
		getAccesoBaseDatos().presentaResponsables(codigo_
		proyecto);
		categories = new String[lista.size()];
		Iterator iterator = lista.iterator();

		while (iterator.hasNext())
		{
		Alarma g = (Alarma) iterator.next();
		categories[i] = "Grupo" +
		((g.getCodigo_alarma()==null)?"X":g.getEstado());

		dataset.addValue(g.getCodigo_alarma() ,
		seriesNames[0], categories[i]);
		i++;
		}
		return dataset;
		}

	/** This producer's data is invalidated after 5 seconds. By this
	method the
	 * producer can influence Cewolf's caching behaviour the way
	it wants to.   */

public boolean hasExpired(Map params, Date since)
	{
	      log.debug(getClass().getName() + "hasExpired()");
	      return (System.currentTimeMillis() since.getTime())  >
	5000;
	}

	/** Returns a unique ID for this DatasetProducer  */
public String getProducerId()
	{
	return "PageViewCountData DatasetProducer";
	}

/** Returns a link target for a special data item. */
public String generateLink(Object data, int series, Object category)
	{
	return seriesNames[series];
	}

/**  @see java.lang.Object#finalize()  */
```

```
          protected void finalize() throws Throwable
                    {
                              super.finalize();
                              log.debug(this + " finalized.");
                    }

/**  @see
org.jfree.chart.tooltips.CategoryToolTipGenerator#generateToolTip(Category
Dataset, int, int)  */
public String generateToolTip(CategoryDataset arg0, int series, int arg2)
          {
                    return seriesNames[series];
          }

public Integer getCodigo_proyecto()
          {
                    return codigo_proyecto;
          }

public void setCodigo_proyecto(Integer codigo_proyecto)
          {
                    this.codigo_proyecto = codigo_proyecto;
          }
}
```

1.2.6 Cal_Alarma.java

```
package sgpi.pojo.general;
import java.sql.Timestamp;
import java.util.Date;
public class Alarma
          {
                    public Alarma(){}
                    Integer codigo_alarma;
                    Date fecha_inicio_act;
                    Date fecha_fin_act;
                    Timestamp fecha_mod_act;
                    Timestamp hora_envio;
                    String estado;
                    Mensaje objmensaje;
                    Historial objhistorial;
                    TipoEnvio objtipoenvio;
                    String repeticiones;
                    Servidor objservidor;
```

```
        public Integer getCodigo_alarma()
                {       return codigo_alarma;           }
        public void setCodigo_alarma(Integer codigo_alarma)
                {       this.codigo_alarma = codigo_alarma;             }
        public String getEstado()
                {       return estado; }
        public void setEstado(String estado)
                {       this.estado = estado; }
        public Timestamp getFecha_mod_act()
                {       return fecha_mod_act;           }
        public void setFecha_mod_act(Timestamp fecha_mod_act)
                {       this.fecha_mod_act = fecha_mod_act;             }
        public Timestamp getHora_envio()
                {       return hora_envio;      }
        public void setHora_envio(Timestamp hora_envio)
                {       this.hora_envio = hora_envio;           }
        public Historial getObjhistorial()
                {       return objhistorial;    }
        public void setObjhistorial(Historial objhistorial)
                {       this.objhistorial = objhistorial;}
        public Mensaje getObjmensaje()
                {       return objmensaje;      }
        public void setObjmensaje(Mensaje objmensaje)
                {       this.objmensaje = objmensaje;           }
        public TipoEnvio getObjtipoenvio()
                {       return objtipoenvio;    }
        public void setObjtipoenvio(TipoEnvio objtipoenvio)
                {       this.objtipoenvio = objtipoenvio;       }
        public String getRepeticiones()
                {       return repeticiones;    }
        public void setRepeticiones(String repeticiones)
                {       this.repeticiones = repeticiones;       }
        public Servidor getObjservidor()
                {       return objservidor;     }
        public void setObjservidor(Servidor objservidor)
                {       this.objservidor = objservidor; }
        public Date getFecha_fin_act()
                {       return fecha_fin_act;  }
        public void setFecha_fin_act(Date fecha_fin_act)
                {       this.fecha_fin_act = fecha_fin_act;    }
        public Date getFecha_inicio_act()
                {       return fecha_inicio_act;        }
        public void setFecha_inicio_act(Date fecha_inicio_act)
                {       this.fecha_inicio_act = fecha_inicio_act;       }
}
```

1.2.7 Cal_Copiar.java

```
package sgpi.pojo.general;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;


public class Copiar
	{

		public void copyDirectory(File srcDir, File dstDir) throws
		IOException
			{
			if (srcDir.isDirectory())
			{
				if (!dstDir.exists())
					{
						dstDir.mkdir();
					}
				String[] children = srcDir.list();
				for (int i=0; i<children.length; i++)
					{
					copyDirectory(new File(srcDir,
					children[i]),
					new File(dstDir, children[i]));
					}
			}
			else
			{
				copy(srcDir, dstDir);
			}
			}

		/** Copia un solo archivo
	 *  @param src
	 *  @param dst
	 *  @throws IOException */

	public void copy(File src, File dst) throws IOException
		{
		InputStream in = new FileInputStream(src);
		OutputStream out = new FileOutputStream(dst);
		byte[] buf = new byte[1024];
```

```
int len;
while ((len = in.read(buf)) > 0)
        {
                out.write(buf, 0, len);
        }
in.close();
out.close();
}

public static void main(String path,String nombrearchivo)
        {
                Copiar cp = new Copiar();
                try
                        {
                                cp.copy(new
                                File(path+nombrearchivo),new
                                File("d:\\archivoadjunto\\"+nombre
                                archivo));
                                System.out.print("Copiado con
                                exito");
                        }
                catch(Exception e)
                        {
                                System.out.println(e);
                        }
        }
}
```

1.2.8 CrearPdf.java

```
package sgpi.pojo.general;

import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;

import com.lowagie.text.Document;
import com.lowagie.text.DocumentException;
import com.lowagie.text.PageSize;
import com.lowagie.text.Paragraph;
import com.lowagie.text.pdf.PdfWriter;
public class CreaPdf {
//String [] args
  public static void main (String msg)
   throws FileNotFoundException,
   DocumentException, IOException {
```

```
    Document d = new Document (PageSize.A4);
    FileOutputStream os = new FileOutputStream("/web-inf/etc/sample.pdf");
    PdfWriter.getInstance (d, os);
    d.open ();
    Paragraph p = new Paragraph (msg);
    d.add (p);
    d.close ();
    os.close();
  }

}
```

1.2.9 Enviador.java

```
package sgpi.pojo.general;

import java.sql.Timestamp;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.List;
import org.springframework.orm.hibernate3.support.HibernateDaoSupport;
import sgpi.AccesoBaseDatos;

public class Enviador extends HibernateDaoSupport implements Runnable {

        AccesoBaseDatos accesoDB2;
        Alarma alarma = new Alarma();
        Mensaje mensaje = new Mensaje();
        Responsable responsable = new Responsable();
        Responsable responsable2 = new Responsable();
        Servidor servidor = new Servidor();
        TipoEnvio tipoenvio = new TipoEnvio();
        Historial historial = new Historial();
        Enviador env = new Enviador();
        boolean e = false;
        Timestamp f = new Timestamp(System.currentTimeMillis());
        Calendar cal = Calendar.getInstance();

        void EnviaEmail(Integer codalarma)
                {
                        responsable =
                        getAccesoDB2().getResponsable(codalarma);
```

```
                responsable2 =
                getAccesoDB2().getResponsable(responsable.codigo_
                origen);
                try
                        {
                        EnvioAttachPrincipal.main("127.0.0.1",
                        responsable2.objpersona.nombre_completo,
                        responsable.objpersona.mail,
                        responsable2.objpersona.mail,
                        alarma.objmensaje.descripcion_asunto,
                        alarma.objmensaje.descripcion_mensaje,
                                                "d:\\archivoadjunto\\" +
                        alarma.objmensaje.nombre_archivo);
                        }
                catch (Exception ex)
                        {
                                e = true;
                        }

                if (e)
                        {
                                return;
                        }
                else
                        {
                                GrabaHistorial();
                        }
        }

void EnviaSms(Integer codalarma)
        {
                responsable =
                getAccesoDB2().getResponsable(codalarma);
                responsable2 =
                getAccesoDB2().getResponsable(responsable.codigo_
                origen);
                try
                        {
                                EnvioAttachPrincipal.main("127.0.0.1",
                                responsable2.objpersona.nombre_compl
                                eto,
                responsable.objpersona.mail,
                responsable2.objpersona.mail,
                alarma.objmensaje.descripcion_asunto,
                alarma.objmensaje.descripcion_mensaje,
                                        "d:\\archivoadjunto\\" +
                                        "mensajesms");
                }
```

```
                catch (Exception ex)
                        {
                                e = true;
                        }

                if (e)
                        {
                                return;
                        }
                else
                        {
                                GrabaHistorial();
                        }
        }

void GrabaHistorial()
        {
historial.setCodigo_alarma(alarma.codigo_alarma);
historial.setCodigo_servidor(alarma.objservidor.codigo_servidor);
historial.setCodigo_tipo_envio(alarma.objtipoenvio.codigo_tipo_envio)
;
historial.setCodigo_mensaje(alarma.objmensaje.codigo_mensaje);
historial.setCodigo_respon_alarma(responsable.codigo_respon_alarm
a);
historial.setFecha_inicio(alarma.fecha_inicio_act);
historial.setFecha_final(alarma.fecha_fin_act);
historial.setFecha_mod(alarma.fecha_mod_act);
historial.setDireccion_ip(alarma.objservidor.direccion_ip);
historial.setDescripcion_asunto(alarma.objmensaje.descripcion_asunt
o);
historial.setDescripcion_mensaje(alarma.objmensaje.descripcion_me
nsaje);
historial.setHora_envio(f);
historial.setCodigo_proyecto(responsable.objproyecto.codigo_proyect
o);
historial.setCodigo_persona(responsable.objpersona.codigo_persona)
;
getAccesoDB2().guardaHistorial(historial);

}

void RevisaEnvio()
        {
                if (alarma.objtipoenvio.codigo_tipo_envio.equals("1"))
                        {
                        // manda a enviarse el email y guardar en
                        historial
                        env.EnviaEmail(alarma.codigo_alarma);
```

```
                        // si tiene repeticiones no marcar estado
                        if (alarma.repeticiones.equals(" "))
                                {
                                alarma.setEstado("E");
                                getAccesoDB2().guardaAlarma(alarma);
                                }
                        }
                else
                        {
                                env.EnviaSms(alarma.codigo_alarma);
                                if (alarma.repeticiones.equals(" "))
                                        {
                                        alarma.setEstado("E");
                                getAccesoDB2().guardaAlarma(alarma);
                                        }
                        }
        }

void Trabaja()
        {
                cal.setTime(f);
                f = new Timestamp(cal.getTimeInMillis());
                SimpleDateFormat d = new
                SimpleDateFormat("h:mm");
                SimpleDateFormat g = new
                SimpleDateFormat("dd/MM/yyyy");
                System.out.println("fecha: " + d.format(f));
                System.out.println("fecha3: " + g.format(f));
                List a = getAccesoDB2().getAlarma2();
                java.util.Iterator b = a.iterator();

                while (b.hasNext())
                        {
                        alarma = (Alarma) b.next();
                        if (alarma.estado.equals("P"))
                                {
                                if (alarma.fecha_inicio_act.equals(f))
                                        {
                                                RevisaEnvio();
                                        }
                                else
                                        {
                                if (alarma.repeticiones.equals("1"))
                                        {
                                        cal.add(Calendar.DATE, 1);
                                        f = new
                                        Timestamp(cal.getTimeInMillis());
```

```
                                        if (f.before(alarma.fecha_fin_act) ||
                                        f.equals(alarma.fecha_fin_act))
                                        RevisaEnvio();
                                        else

                                        {

                                        alarma.setEstado("E");
                                        getAccesoDB2().guardaAlarma(alarma);
                                        // ojo para actualizar

                                }
                        }
                                        if (alarma.repeticiones.equals("2"))
                                        {
                                        cal.add(Calendar.DATE, 7);
                                        f = new Timestamp(cal.getTimeInMillis());
                                        if (f.before(alarma.fecha_fin_act) ||
                                        f.equals(alarma.fecha_fin_act))
                                        RevisaEnvio();
                                        else
                                        {
                                        alarma.setEstado("E");
                                        getAccesoDB2().guardaAlarma(alarma);
                                        // ojo para actualizar

                                        }
                                }
                        }
                }
        }
}


public static void main()
        {
        Enviador env2 = new Enviador();
        env2.Trabaja();
        }

public AccesoBaseDatos getAccesoDB2()
        {
                return accesoDB2;
        }

public void setAccesoDB2(AccesoBaseDatos accesoDB2)
        {
                this.accesoDB2 = accesoDB2;
```

```
        }

public void run() { }


private static int count  = 0;
public void job()
        {
                System.out.println("hola soy camilo: "+ count++);
        }
}
```

1.2.10 EnvioAttachPrincipal.java

```
package sgpi.pojo.general;

import java.io.IOException;
import java.util.Date;
import java.util.Properties;
import javax.mail.Address;
import javax.mail.Message;
import javax.mail.MessagingException;
import javax.mail.Multipart;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeBodyPart;
import javax.mail.internet.MimeMessage;
import javax.mail.internet.MimeMultipart;



public class EnvioAttachPrincipal
        {
                public static void main(String host,String from,String to,String
cc,String asunto,String mensaje,String RutaArchivoadjunto)
                        {
                                Properties props = new Properties();
                                props.put("mail.smtp.auth", host);
                                /*  Session es la clase base, define una sesion
                                de correo, se crea una unica sesion
                                con el método getDefaultInstance()la sesion
                                puede ser compartida.          */
                                Session session = Session.getInstance( props,
                                null);
```

```
/*  Crear el mensaje que quiero enviar. El objeto
MimeMessage permite
enviar más que texto plano...imágenes... */

MimeMessage message = new
MimeMessage(session);
try
        {
        // Rellenar el texto (sólo texto plano).
        // message.setText("Probando JavaMail
        1.4");
        message.setSubject(from);
        message.setSentDate(new Date());
        MimeBodyPart mbp1 = new
        MimeBodyPart(); // creamos y
        rellenamos la primera parte del mensaje.
        mbp1.setText(mensaje);
        MimeBodyPart mbp2 = new
        MimeBodyPart();  // creamos y
        rellenamos la segunda parte del
        mensaje.
        try
        {
        mbp2.attachFile(RutaArchivoadjunto); //
        adjuntamos un archivo.
        }
           catch (IOException e)
                {
                e.printStackTrace();
                }
           Multipart mp = new MimeMultipart();  //
           creamos el objeto multipart.
           mp.addBodyPart(mbp1);
           mp.addBodyPart(mbp2);
           message.setContent(mp); // añadimos
           el objeto al mensaje.
           //son necesarios objs address para el
           from y el to
          //Indicar quien lo envia
         Address Fromaddress;
         Fromaddress = new InternetAddress
         (from);
         message.setFrom(Fromaddress);
         //conectamos la direccion de FROM al
         objeto message
          Address TOaddress = new
         InternetAddress  //Indicar a donde lo
         envias
```

```
                                        (to);

                                        message.addRecipient(Message.Recipie
                                        ntType.TO,TOaddress); //conectamos la
                                        direccion de TO al objeto message
                                        Address CCaddress = new
                                        InternetAddress
                                        (cc); //envio de copia del mensaje

                                        message.addRecipient(Message.Recipie
                                        ntType.CC,CCaddress);
                                        message.saveChanges();// implícito con
                                        send()
                                        Transport transport =
                                        session.getTransport("smtp");
                                        Transport.send(message);
                                        transport.close();
                                        System.out.println("Mensaje enviado
                                        correctamente");
                                        }
                                catch (MessagingException mexec)
                                {
                                        mexec.printStackTrace();
                                }
                        }
        }
```

1.2.11 Cal_Examplejob.java

```
package sgpi.pojo.general;g

import java.sql.Timestamp;
import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.List;

import org.springframework.orm.hibernate3.support.HibernateDaoSupport;
```

```
public class ExampleJob extends HibernateDaoSupport /*extends
QuartzJobBean*/
	{

		public ExampleJob() {}
		private static int count = 0;
		private int timeout;
		**  Setter called after the ExampleJob is instantiated with the
		value from
		* the JobDetailBean (5)  */
		public void setTimeout(int timeout)
			{
				this.timeout = timeout;
			}
		public void guardaHistorial(Historial his)
			{
				getHibernateTemplate().saveOrUpdate(his);
			}
		public void guardaResponsable(Responsable l)
			{
				getHibernateTemplate().saveOrUpdate(l);
			}
		public void guardaAlarma(Alarma al)
			{
				getHibernateTemplate().saveOrUpdate(al);
			}

		public void EnviaEmail(Responsable responsable,Timestamp
		hSistema,Date fsistema)
			{
			boolean e = false;
			String mailfrom = null;
			//lee el codigo y email de quien envia
			if (responsable.getCodigo_origen().longValue()>0)
				{
				List lst =  getHibernateTemplate().find("from
				Persona per where per.codigo_persona =
				?",responsable.getCodigo_origen());
					if (lst != null && lst.size() >0 )
					{
					Persona per = (Persona) lst.get(0);
					mailfrom = per.getMail();
					}
					String mail =
					responsable.getObjpersona().getMail();
					if (mailfrom==null || mail==null)
					{
```

```
System.out.println("Origen o Destino no
Poseen Email");
return;
}
String archivo =
responsable.getObjalarma().getObjmensaje().ge
tNombre_archivo();
if (mailfrom.length()>0 && mail.length()>0 )
{
if (archivo == null)
{
try
{
Sendmail2.main(responsable.getObjalarma().getObjservidor().getDire
ccion_ip(),mailfrom,mail,
responsable.getObjalarma().getObjmensaje().getDescripcion_asunto()
,
responsable.getObjalarma().getObjmensaje().getDescripcion_mensaj
e());

}

catch (Exception ex)
{
e = true;
}
}
else
{
try
{
EnvioAttachPrincipal.main(responsable.getObjalarma().getObjservidor
().getDireccion_ip(),mailfrom,mail,mailfrom,
responsable.getObjalarma().getObjmensaje().getDescripcion_asunto()
,
responsable.getObjalarma().getObjmensaje().getDescripcion_mensaj
e(),
"d:\\archivoadjunto\\" + archivo);
}
catch (Exception ex)
{
e = true;
}
}
}
}
if (e)
{
```

```
        System.out.println("MENSAJE NO ENVIADO...FALTA ALGUN
        PARAMETRO O EMAIL INCORRECTO");
        return;
        }
        else
        {
        GrabaHistorial(responsable,hSistema,fsistema);
        }
}

public void EnviaSms(Responsable responsable,Timestamp hSistema,Date
fsistema)
        {
        boolean e = false;
        String mailfrom = "";
        if (responsable.getCodigo_origen().longValue()>0)
        {
        List lst =  getHibernateTemplate().find("from Persona per where
        per.codigo_persona = ?",responsable.getCodigo_origen());
                if (lst != null && lst.size() >0 )
                {
                        Persona per = (Persona) lst.get(0);
                        mailfrom = per.getMail();
                }
                  String mail = responsable.getObjpersona().getMail();
                  String archivo =
                  responsable.getObjalarma().getObjmensaje().getNombre_arc
                  hivo();
                  if (archivo == null)
                        {
                                archivo = "MENSAJESMS.TXT";
                        }
                  if (mailfrom.length()>0 && mail.length()>0 )
                        {
                        try
                        {
        EnvioAttachPrincipal.main("172.20.6.18",mailfrom,mail,mailfrom,
        responsable.getObjalarma().getObjmensaje().getDescripcion_asunto()
        ,
        responsable.getObjalarma().getObjmensaje().getDescripcion_mensaj
        e(),  "d:\\archivoadjunto\\" + archivo);
        }
        catch (Exception ex)
                {
                        e = true;
                }
        }
}
```

```
if (e)
        {
        System.out.println("Mensaje NO Enviado");
        return;
        }
else
        {
        GrabaHistorial(responsable,hSistema,fsistema);
        }
        }



public void GrabaHistorial(Responsable responsable,Timestamp
hSistema,Date fsistema)
        {
        Historial historial = new Historial();
        historial.setCodigo_alarma(responsable.getObjalarma().getCodigo_al
        arma());
        historial.setCodigo_servidor(responsable.getObjalarma().getObjservid
        or().getCodigo_servidor());
        historial.setCodigo_tipo_envio(responsable.getObjalarma().getObjtipo
        envio().getCodigo_tipo_envio());
        historial.setCodigo_mensaje(responsable.getObjalarma().getObjmens
        aje().getCodigo_mensaje());
        historial.setCodigo_respon_alarma(responsable.getCodigo_respon_al
        arma());
        historial.setFecha_inicio(responsable.getObjalarma().getFecha_inicio
        _act());
        historial.setFecha_final(responsable.getObjalarma().getFecha_fin_act
        ());
        historial.setFecha_mod(responsable.getObjalarma().getFecha_mod_a
        ct());
        historial.setDireccion_ip(responsable.getObjalarma().getObjservidor().
        getDireccion_ip());
        historial.setDescripcion_asunto(responsable.getObjalarma().getObjme
        nsaje().getDescripcion_asunto());
        historial.setDescripcion_mensaje(responsable.getObjalarma().getObj
        mensaje().getDescripcion_mensaje());
        historial.setHora_envio(hSistema);
        historial.setCodigo_proyecto(responsable.getObjproyecto().getCodigo
        _proyecto());
        historial.setCodigo_persona(responsable.getObjpersona().getCodigo_
        persona());
        historial.setFecha_mod(fsistema);
        guardaHistorial(historial);
```

```
List lstalarma =  getHibernateTemplate().find("from Alarma al where
al.codigo_alarma =
?",responsable.getObjalarma().getCodigo_alarma());

        if (lstalarma != null && lstalarma.size() >0 )
                {
                        Alarma alar = (Alarma) lstalarma.get(0);
                        alar.setHora_envio(hSistema);
                        guardaAlarma(alar);
                }
        }

public void RevisaEnvio(Responsable responsable, Timestamp
hSistema, Date fsistema)
        {
        if
        (responsable.getObjalarma().getObjtipoenvio().getCodigo_tipo
        _envio().equals(1))
        {
        EnviaEmail(responsable,hSistema,fsistema);   // manda a
        enviarse el email y guardar en historial
        // si no tiene repeticiones marcar estado con E (enviado) pilas
        if (responsable.getObjalarma().getRepeticiones().equals(" "))
                {
                responsable.getObjalarma().setEstado("E");
                guardaResponsable(responsable);
                }
        }
        else
        {
        EnviaSms(responsable,hSistema,fsistema);
        if (responsable.getObjalarma().getRepeticiones().equals(" "))
                {
                responsable.getObjalarma().setEstado("E");
                guardaResponsable(responsable);
                }
        }
}

public void doIt()
{
        System.out.println("Proceso de Hilo No: " + count++);
        List list =  getHibernateTemplate().find(
                 "from Responsable  rs where rs.objalarma.estado='P'
                order by rs.codigo_respon_alarma");
        if (list != null && list.size() >0 )
                {
```

```
Responsable responsable2 = (Responsable)
list.get(0);
  for (java.util.Iterator b = list.iterator(); b.hasNext();)
      {
        Responsable responsable = (Responsable)
        b.next();
        Calendar cal = Calendar.getInstance();

      cal.setTime(responsable.objalarma.getFecha_in
      icio_act());   // Set it in the Calendar object
        cal.add(Calendar.DATE, 1);       // Add 1 days
        Date fregistro = cal.getTime();
        System.out.println("fe  "+fregistro);

        DateFormat defaultDate =
        DateFormat.getDateInstance(3);
        String f=defaultDate.format(fregistro);

        Calendar cal2 = Calendar.getInstance();
        cal2.setTime(new Date());  // Fecha Actual
        para comparacion tipo Date
        Date fwindows = cal2.getTime();
        String g=defaultDate.format(fwindows);
        System.out.println("fi
        "+f+g+"exp"+fregistro+fwindows );
        System.out.println("codigos
        "+responsable.getCodigo_respon_alarma() );

        Calendar cal3 = Calendar.getInstance();

      cal.setTime(responsable.objalarma.getFecha_in
      icio_act());
       cal.add(Calendar.DATE, 7);        // Add 7 days
        Date fsemanal = cal.getTime();
        String h=defaultDate.format(fsemanal);

        Calendar cal4 = Calendar.getInstance();

      cal.setTime(responsable.objalarma.getFecha_in
      icio_act());
       Date finicio = cal.getTime();
       String i=defaultDate.format(finicio);

        Calendar cal5 = Calendar.getInstance();
cal.setTime(responsable.objalarma.getFecha_fin_act());
        Date ffinal = cal.getTime();
        String j=defaultDate.format(ffinal);
        String k = "";
```

```
Timestamp fecmodif = responsable.objalarma.getHora_envio();

if (fecmodif==null){}
else{
Calendar cal6 = Calendar.getInstance();
cal.setTime(responsable.objalarma.getHora_envio());
Date fmod = cal.getTime();
k=defaultDate.format(fmod);
}
java.util.Date utilDate = new java.util.Date(); //fecha actual en tipo
timestamp
long lnMilisegundos = utilDate.getTime();
java.sql.Timestamp sqlTimestamp = new
java.sql.Timestamp(lnMilisegundos);

SimpleDateFormat s,t;
String texto;
s = new SimpleDateFormat("dd/MM/yyyy");
Timestamp fDiaria= null;
Timestamp fSistema = null;
Timestamp fSemanal = null;
Timestamp fInicioActiv = null;
Timestamp fFinActiv = null;
Timestamp fModact = null;
try {
        fDiaria = new Timestamp(s.parse(f).getTime());
        fSistema = new Timestamp(s.parse(g).getTime());
        fSemanal = new Timestamp(s.parse(h).getTime());
        fInicioActiv = new Timestamp(s.parse(i).getTime());
        fFinActiv = new Timestamp(s.parse(j).getTime());
        if (fecmodif==null){}else
        fModact = new Timestamp(s.parse(k).getTime());
        }
catch (ParseException e)
        {
           e.printStackTrace();
        }
if (fInicioActiv.equals(fSistema) && fecmodif==null )
RevisaEnvio(responsable,sqlTimestamp,fSistema);
else
        {
        if (responsable.getObjalarma().getRepeticiones().equals("1"))
        ((fDiaria.after(fInicioActiv) && fDiaria.before(fFinActiv)) ||
        (fDiaria.equals(fFinActiv) || fDiaria.equals(fInicioActiv)))
       {
       if (fecmodif==null)
       {
```

```
        RevisaEnvio(responsable,sqlTimestamp,fSistema);
        }
        else
        {

        if (fModact.equals(fSistema)){}
        else RevisaEnvio(responsable,sqlTimestamp,fSistema);
        }
        }
        else
         {
        if (fDiaria.after(fFinActiv))
        {
        List lstalarmaE =  getHibernateTemplate().find("from Alarma al
        where al.codigo_alarma =
        ?",responsable.getObjalarma().getCodigo_alarma());

        if (lstalarmaE != null && lstalarmaE.size() >0 )
        {
        Alarma alar = (Alarma) lstalarmaE.get(0);
        alar.setEstado("E");
        guardaAlarma(alar);
        }

}
}
}
if (responsable.getObjalarma().getRepeticiones().equals("2"))
{
if ((fSemanal.after(fInicioActiv) && fSemanal.before(fFinActiv)) ||
   (fSemanal.equals(fFinActiv) || fSemanal.equals(fInicioActiv)))
RevisaEnvio(responsable,sqlTimestamp,fSistema);
else
{
        if (fSemanal.after(fFinActiv))
        {
        List lstalarmaE =  getHibernateTemplate().find("from Alarma al
        where al.codigo_alarma =
        ?",responsable.getObjalarma().getCodigo_alarma());

        if (lstalarmaE != null && lstalarmaE.size() >0 )
        {
        Alarma alar = (Alarma) lstalarmaE.get(0);
                alar.setEstado("E");
                guardaAlarma(alar);
        }
        }
```

```
                }
            }
        }
    }
}
```

1.2.12 Cal_Historial.java(Paquete sgpi.pijo.general)

```
package sgpi.pojo.general;
import java.sql.Timestamp;
import java.util.Date;


public class Cal_Historial {

	public Cal_Historial(){

	}

	Integer codigo_historial;
	Integer codigo_alarma;
	Integer codigo_servidor;
	Integer codigo_tipo_envio;
	Integer codigo_mensaje;
	Integer codigo_respon_alarma;
	Date fecha_inicio;
	Date fecha_final;
	Date fecha_mod;
	Timestamp hora_envio;
	Integer codigo_actividad;
	String direccion_ip;
	String descripcion_asunto;
	String descripcion_mensaje;

	Integer codigo_proyecto;
	Integer codigo_persona;
	Cal_Alarma objalarma;

	public Integer getCodigo_actividad() {
		return codigo_actividad;
	}
	public void setCodigo_actividad(Integer codigo_actividad) {
		this.codigo_actividad = codigo_actividad;
	}
	public Integer getCodigo_alarma() {
```

```
		return codigo_alarma;
	}
	public void setCodigo_alarma(Integer codigo_alarma) {
		this.codigo_alarma = codigo_alarma;
	}
	public Integer getCodigo_historial() {
		return codigo_historial;
	}
	public void setCodigo_historial(Integer codigo_historial) {
		this.codigo_historial = codigo_historial;
	}
	public Integer getCodigo_mensaje() {
		return codigo_mensaje;
	}
	public void setCodigo_mensaje(Integer codigo_mensaje) {
		this.codigo_mensaje = codigo_mensaje;
	}
	public Integer getCodigo_persona() {
		return codigo_persona;
	}
	public void setCodigo_persona(Integer codigo_persona) {
		this.codigo_persona = codigo_persona;
	}
	public Integer getCodigo_proyecto() {
		return codigo_proyecto;
	}
	public void setCodigo_proyecto(Integer codigo_proyecto) {
		this.codigo_proyecto = codigo_proyecto;
	}
	public Integer getCodigo_respon_alarma() {
		return codigo_respon_alarma;
	}
	public void setCodigo_respon_alarma(Integer codigo_respon_alarma)
{
		this.codigo_respon_alarma = codigo_respon_alarma;
	}
	public Integer getCodigo_servidor() {
		return codigo_servidor;
	}
	public void setCodigo_servidor(Integer codigo_servidor) {
		this.codigo_servidor = codigo_servidor;
	}
	public Integer getCodigo_tipo_envio() {
		return codigo_tipo_envio;
	}
	public void setCodigo_tipo_envio(Integer codigo_tipo_envio) {
		this.codigo_tipo_envio = codigo_tipo_envio;
	}
```

```
public String getDescripcion_asunto() {
	return descripcion_asunto;
}
public void setDescripcion_asunto(String descripcion_asunto) {
	this.descripcion_asunto = descripcion_asunto;
}
public String getDescripcion_mensaje() {
	return descripcion_mensaje;
}
public void setDescripcion_mensaje(String descripcion_mensaje) {
	this.descripcion_mensaje = descripcion_mensaje;
}
public String getDireccion_ip() {
	return direccion_ip;
}
public void setDireccion_ip(String direccion_ip) {
	this.direccion_ip = direccion_ip;
}

public Cal_Alarma getObjalarma() {
	return objalarma;
}
public void setObjalarma(Cal_Alarma objalarma) {
	this.objalarma = objalarma;
}


public Timestamp getHora_envio() {
	return hora_envio;
}
public void setHora_envio(Timestamp hora_envio) {
	this.hora_envio = hora_envio;
}
public Date getFecha_final() {
	return fecha_final;
}
public void setFecha_final(Date fecha_final) {
	this.fecha_final = fecha_final;
}
public Date getFecha_inicio() {
	return fecha_inicio;
}
public void setFecha_inicio(Date fecha_inicio) {
	this.fecha_inicio = fecha_inicio;
}
public Date getFecha_mod() {
	return fecha_mod;
}
```

```
        public void setFecha_mod(Date fecha_mod) {
                this.fecha_mod = fecha_mod;
        }
}
```

1.2.13 Cal_mensaje.java (Paquete Sgpi.pojo.general)

```
package sgpi.pojo.general;

import java.io.Serializable;
import java.sql.Blob;
import java.sql.SQLException;

import org.hibernate.lob.BlobImpl;

public class Cal_Mensaje{

        public Cal_Mensaje(){

        }

        Integer codigo_mensaje;
        String descripcion_mensaje;
        String descripcion_asunto;
        String ruta_archivo;
        String nombre_archivo;
        Blob archivo_adj;
    byte[] archivo_adjunto;

        public byte[] getArchivo_adjunto() {

                if (getArchivo_adj() != null) {
                        try {
                                archivo_adjunto = getArchivo_adj().getBytes(1L,
(int)getArchivo_adj().length());
                        } catch (SQLException e) {
                                e.printStackTrace();
                        }
                }

                return archivo_adjunto;
        }

        public void setArchivo_adjunto(byte[] archivo_adjunto) {
                this.archivo_adjunto = archivo_adjunto;
```

```
        BlobImpl b = new BlobImpl(getArchivo_adjunto());

        setArchivo_adj(b);
}

public Integer getCodigo_mensaje() {
        return codigo_mensaje;
}

public void setCodigo_mensaje(Integer codigo_mensaje) {
        this.codigo_mensaje = codigo_mensaje;
}

public String getDescripcion_asunto() {
        return descripcion_asunto;
}

public void setDescripcion_asunto(String descripcion_asunto) {
        this.descripcion_asunto = descripcion_asunto;
}

public String getDescripcion_mensaje() {
        return descripcion_mensaje;
}

public void setDescripcion_mensaje(String descripcion_mensaje) {
        this.descripcion_mensaje = descripcion_mensaje;
}

public Blob getArchivo_adj() {
        return archivo_adj;
}

public void setArchivo_adj(Blob archivo_adj) {
        this.archivo_adj = archivo_adj;
}

public String getNombre_archivo() {
        return nombre_archivo;
}

public void setNombre_archivo(String nombre_archivo) {
        this.nombre_archivo = nombre_archivo;
}

public String getRuta_archivo() {
        return ruta_archivo;
}
```

```
	public void setRuta_archivo(String ruta_archivo) {
		this.ruta_archivo = ruta_archivo;
	}
}
```

1.2.14 Cal_participante.java (Paquete sgpi.pojo.general)

```
package sgpi.pojo.general;

public class Cal_Participante {

	public Cal_Participante(){

	}
	Integer codigo_participante;
	String descripcion;
	Cal_Persona objpersona;

	public Cal_Persona getObjpersona() {
		return objpersona;
	}
	public void setObjpersona(Cal_Persona objpersona) {
		this.objpersona = objpersona;
	}
	public Integer getCodigo_participante() {
		return codigo_participante;
	}
	public void setCodigo_participante(Integer codigo_participante) {
		this.codigo_participante = codigo_participante;
	}
	public String getDescripcion() {
		return descripcion;
	}
	public void setDescripcion(String descripcion) {
		this.descripcion = descripcion;
	}

}
```

1.2.15 Cal_persona.java (Paquete sgpi.pojo.general)

```
package sgpi.pojo.general;

public class Cal_Persona {

        public Cal_Persona(){

        }

        Integer codigo_persona;
        String nombre_completo;
        String telefono;
        String mail;


        public Integer getCodigo_persona() {
                return codigo_persona;
        }
        public void setCodigo_persona(Integer codigo_persona) {
                this.codigo_persona = codigo_persona;
        }
        public String getNombre_completo() {
                return nombre_completo;
        }
        public void setNombre_completo(String nombre_completo) {
                this.nombre_completo = nombre_completo;
        }
        public String getMail() {
                return mail;
        }
        public void setMail(String mail) {
                this.mail = mail;
        }
        public String getTelefono() {
                return telefono;
        }
        public void setTelefono(String telefono) {
                this.telefono = telefono;
        }

}
```

1.2.16 Cal_proyecto.java (Paquete sgpi.pojo.general)

```
package sgpi.pojo.general;

import java.sql.Date;
import java.util.Set;

public class Cal_Proyecto {

	public Cal_Proyecto(){

	}
	Integer codigo_proyecto;
	String nombre_proyecto;
	Date fecha_inicio_real;
	Date fecha_fin_real;
	Integer codigo_responsable;

	Set participantes;



	public Set getParticipantes() {
		return participantes;
	}
	public void setParticipantes(Set participantes) {
		this.participantes = participantes;
	}
	public Integer getCodigo_proyecto() {
		return codigo_proyecto;
	}
	public void setCodigo_proyecto(Integer codigo_proyecto) {
		this.codigo_proyecto = codigo_proyecto;
	}
	public Integer getCodigo_responsable() {
		return codigo_responsable;
	}
	public void setCodigo_responsable(Integer codigo_responsable) {
		this.codigo_responsable = codigo_responsable;
	}
	public Date getFecha_fin_real() {
		return fecha_fin_real;
	}
	public void setFecha_fin_real(Date fecha_fin_real) {
		this.fecha_fin_real = fecha_fin_real;
	}
	public Date getFecha_inicio_real() {
		return fecha_inicio_real;
```

```
	}
	public void setFecha_inicio_real(Date fecha_inicio_real) {
		this.fecha_inicio_real = fecha_inicio_real;
	}
	public String getNombre_proyecto() {
		return nombre_proyecto;
	}
	public void setNombre_proyecto(String nombre_proyecto) {
		this.nombre_proyecto = nombre_proyecto;
	}
}
```

1.2.17 Cal_Responsable.java (Paquete sgpi.pojo.general)

```
package sgpi.pojo.general;

public class Cal_Responsable {

	public Cal_Responsable(){

	}

	Cal_Proyecto objproyecto;
	Integer codigo_respon_alarma;
	Integer codigo_origen;
	Cal_Alarma objalarma;
	Cal_Persona objpersona;

	public Integer getCodigo_origen() {
		return codigo_origen;
	}
	public void setCodigo_origen(Integer codigo_origen) {
		this.codigo_origen = codigo_origen;
	}

	public Integer getCodigo_respon_alarma() {
		return codigo_respon_alarma;
	}
	public void setCodigo_respon_alarma(Integer codigo_respon_alarma)
	{
```

```
		this.codigo_respon_alarma = codigo_respon_alarma;
	}
	public Cal_Alarma getObjalarma() {
		return objalarma;
	}
	public void setObjalarma(Cal_Alarma objalarma) {
		this.objalarma = objalarma;
	}

	public Cal_Persona getObjpersona() {
		return objpersona;
	}
	public void setObjpersona(Cal_Persona objpersona) {
		this.objpersona = objpersona;
	}
	public Cal_Proyecto getObjproyecto() {
		return objproyecto;
	}
	public void setObjproyecto(Cal_Proyecto objproyecto) {
		this.objproyecto = objproyecto;
	}


}
```

1.2.18 Cal_sendMail3.java (Paquete sgpi.pojo.general)

```
package sgpi.pojo.general;

import java.util.Properties;
import javax.mail.Message;
import javax.mail.Session;
import javax.mail.Transport;
import javax.mail.internet.InternetAddress;
import javax.mail.internet.MimeMessage;


public class Cal_Sendmail2 {
	public static void main(String host,String from,String to,String
	mensaje,String texto) {
	 boolean err = false;
		Properties prop = System.getProperties();
		prop.put("mail.smtp.host",host);
		Session ses1 = Session.getDefaultInstance(prop,null);
		MimeMessage msg = new MimeMessage(ses1);
		try{
```

```
				msg.setFrom(new InternetAddress(from));
				msg.addRecipient(Message.RecipientType.TO, new
						InternetAddress(to));
				msg.setSubject(mensaje);
				msg.setText(texto);
				Transport.send(msg);
				}catch (Exception e) {
					err=true;
				}

				if (err){
					System.out.println("Mensaje NO Enviado");
					return;
				} else {
					System.out.println("Mensaje Enviado Exitosamente");
					return;
				}

		}

}
```

1.2.19 Cal_Servidor.java (Paquete sgpi.pojo.general)

```
package sgpi.pojo.general;

public class Cal_Servidor {

	public Cal_Servidor(){

	}

	Integer codigo_servidor;
	String nombre_servidor;
	String codigo_usuario;
	String direccion_ip;
	String dominio;
	String puerto;


	public Integer getCodigo_servidor() {
		return codigo_servidor;
	}
	public void setCodigo_servidor(Integer codigo_servidor) {
		this.codigo_servidor = codigo_servidor;
```

```
	}
	public String getCodigo_usuario() {
		return codigo_usuario;
	}
	public void setCodigo_usuario(String codigo_usuario) {
		this.codigo_usuario = codigo_usuario;
	}
	public String getDireccion_ip() {
		return direccion_ip;
	}
	public void setDireccion_ip(String direccion_ip) {
		this.direccion_ip = direccion_ip;
	}
	public String getDominio() {
		return dominio;
	}
	public void setDominio(String dominio) {
		this.dominio = dominio;
	}
	public String getNombre_servidor() {
		return nombre_servidor;
	}
	public void setNombre_servidor(String nombre_servidor) {
		this.nombre_servidor = nombre_servidor;
	}
	public String getPuerto() {
		return puerto;
	}
	public void setPuerto(String puerto) {
		this.puerto = puerto;
	}
}
```

1.2.20 Cal_Tipo_Envio.java (Paquete sgpi.pojo.general)

```
package sgpi.pojo.general;

public class Cal_TipoEnvio {
	Integer codigo_tipo_envio;
	String desc_envio;

	public Cal_TipoEnvio() {

	}
```

```
        public Integer getCodigo_tipo_envio() {
                return codigo_tipo_envio;
        }

        public void setCodigo_tipo_envio(Integer codigo_tipo_envio) {
                this.codigo_tipo_envio = codigo_tipo_envio;
        }

        public String getDesc_envio() {
                return desc_envio;
        }

        public void setDesc_envio(String desc_envio) {
                this.desc_envio = desc_envio;
        }
}
```

1.2.21 Mapping_Cal.htm.xml (Paquete sgpi.pojo.general)

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping
DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping>

   <class name="sgpi.pojo.general.Cal_Participante"
   table="CLC_PARTICIPANTE">
      <id name="codigo_participante" type="int">
         <column name="codigo_participante" />
      </id>

   <property name="descripcion" type="string">
         <column name="descripcion" />
   </property>

     <many-to-one name="objpersona"
               class="sgpi.pojo.general.Cal_Persona"
               lazy="false" fetch="select" not-found="ignore" update="false">
        <column name="codigo_persona"/>
      </many-to-one>
   </class>
```

```
 <class name="sgpi.pojo.general.Cal_TipoEnvio" table="cal_tipo_envio">
      <id name="codigo_tipo_envio" type="int">
         <column name="codigo_tipo_envio" />
      </id>

      <property name="desc_envio" type="string">
         <column name="desc_envio" />
      </property>
      </class>


    <class name="sgpi.pojo.general.Cal_Servidor" table="cal_servidor">
         <id name="codigo_servidor" type="int">
         <column name="codigo_servidor" />
         <generator class="sequence">
                        <param name="sequence">cal_seq_servidor</param>
                        </generator>
      </id>

      <property name="nombre_servidor" type="string">
         <column name="nombre_servidor" />
      </property>
      <property name="codigo_usuario" type="string">
         <column name="codigo_usuario" />
      </property>
               <property name="direccion_ip" type="string">
         <column name="direccion_ip" />
      </property>
               <property name="dominio" type="string">
         <column name="dominio" />
      </property>
               <property name="puerto" type="string">
         <column name="puerto" />
      </property>
   </class>


   <class name="sgpi.pojo.general.Cal_Proyecto"
table="GEN_PROYECTOS">

      <id name="codigo_proyecto" type="int">
         <column name="codigo_proyecto" />
      </id>

      <property name="nombre_proyecto" type="string">
         <column name="nombre_proyecto" />
      </property>
```

```
  <set name="participantes" inverse="true" lazy="false"
       table="clc_participante_proyecto"
       fetch="select" cascade="all-delete-orphan">
      <key>
         <column name="codigo_proyecto"/>
      </key>
      <many-to-many column="codigo_participante"
      class="sgpi.pojo.general.Cal_Participante" />
  </set>

 </class>

 <class name="sgpi.pojo.general.Cal_Persona" table="GEN_personas" >

  <id name="codigo_persona" type="int">
      <column name="codigo_persona" />
  </id>

  <property name="nombre_completo" type="string">
      <column name="nombre_completo" />
  </property>
 <property name="telefono" type="string">
      <column name="telefono" />
  </property>
  <property name="mail" type="string">
      <column name="mail" />
  </property>
 </class>



<class name="sgpi.pojo.general.Cal_Mensaje" table="cal_mensaje" >
  <id name="codigo_mensaje" type="int">
      <column name="codigo_mensaje" />
      <generator class="sequence">
                      <param name="sequence">cal_seq_mensaje</param>
                      </generator>
  </id>

  <property name="descripcion_mensaje" type="string">
      <column name="descripcion_mensaje" />
  </property>
  <property name="descripcion_asunto" type="string">
      <column name="descripcion_asunto" />
  </property>

  <property name="archivo_adj" type="blob">
      <column name="archivo_adjunto" />
```

```
    </property>
    <property name="ruta_archivo" type="string">
       <column name="ruta_archivo" />
    </property>
    <property name="nombre_archivo" type="string">
       <column name="nombre_archivo" />
    </property>
  </class>

 <class name="sgpi.pojo.general.Cal_Historial"
 <table="cal_historial_mensaje">
    <id name="codigo_historial" type="int">
       <column name="codigo_historial" />
       <generator class="sequence">
                        <param name="sequence">cal_seq_historial</param>
                        </generator>
    </id>
    <property name="codigo_alarma" type="integer">
       <column name="codigo_alarma" />
    </property>
    <property name="codigo_servidor" type="integer">
       <column name="codigo_servidor" />
    </property>

    <property name="codigo_tipo_envio" type="integer">
       <column name="codigo_tipo_envio" />
    </property>
    <property name="codigo_mensaje" type="integer">
       <column name="codigo_mensaje" />
    </property>
    <property name="codigo_respon_alarma" type="integer">
       <column name="codigo_respon_alarma" />
    </property>
    <property name="fecha_inicio" type="date">
       <column name="fecha_inicio" />
    </property>
    <property name="fecha_final" type="date">
       <column name="fecha_final" />
    </property>
    <property name="fecha_mod" type="date">
       <column name="fecha_mod" />
    </property>
   <property name="codigo_actividad" type="integer">
       <column name="codigo_actividad" />
    </property>
    <property name="direccion_ip" type="string">
       <column name="direccion_ip" />
    </property>
```

```
        <property name="descripcion_asunto" type="string">
          <column name="descripcion_asunto" />
        </property>
        <property name="descripcion_mensaje" type="string">
          <column name="descripcion_mensaje" />
        </property>
        <property name="hora_envio" type="timestamp">
          <column name="hora_envio" />
        </property>
        <property name="codigo_proyecto" type="integer">
          <column name="codigo_proyecto" />
        </property>
        <property name="codigo_persona" type="integer">
          <column name="codigo_persona" />
        </property>
    </class>


    <class name="sgpi.pojo.general.Cal_Responsable"
    table="cal_responsable_alarma" >

        <id name="codigo_respon_alarma" type="int">
          <column name="codigo_respon_alarma" />
          <generator class="sequence">
                              <param
name="sequence">cal_seq_responsable</param>
                              </generator>
        </id>

      <property name="codigo_origen" type="integer">
          <column name="codigo_origen" />
        </property>

          <many-to-one name="objalarma"
                  class="sgpi.pojo.general.Cal_Alarma"
                  lazy="false" fetch="select" not-found="ignore" update="false">
          <column name="codigo_alarma"/>
         </many-to-one>
        <many-to-one name="objproyecto"
                  class="sgpi.pojo.general.Cal_Proyecto"
                  lazy="false" fetch="select" not-found="ignore" update="false">
          <column name="codigo_proyecto"/>
         </many-to-one>
          <many-to-one name="objpersona"
            class="sgpi.pojo.general.Cal_Persona"
            lazy="false" fetch="select" not-found="ignore" update="false">
          <column name="codigo_persona"/>
          </many-to-one>
```

```
 </class>


<class name="sgpi.pojo.general.Cal_Alarma" table="cal_alarma" >

  <id name="codigo_alarma" type="int">
     <column name="codigo_alarma" />
     <generator class="sequence">
                 <param name="sequence">cal_seq_alarma</param>
                 </generator>
  </id>

  <property name="fecha_inicio_act" type="date">
     <column name="fecha_inicio_act" />
  </property>
  <property name="fecha_fin_act" type="date">
     <column name="fecha_fin_act" />
  </property>

  <property name="fecha_mod_act" type="timestamp">
     <column name="fecha_mod_act" />
  </property>
  <property name="hora_envio" type="timestamp">
     <column name="hora_envio" />
  </property>
  <property name="estado" type="string">
     <column name="estado" />
  </property>
   <property name="repeticiones" type="string">
     <column name="repeticiones" />
  </property>


 <many-to-one name="objmensaje"
 class="sgpi.pojo.general.Cal_Mensaje" lazy="false">
    <column name="codigo_mensaje"/>
  </many-to-one>

  <many-to-one name="objtipoenvio"
  class="sgpi.pojo.general.Cal_TipoEnvio" lazy="false">
    <column name="codigo_tipo_envio"/>
  </many-to-one>
    <many-to-one name="objservidor"
    class="sgpi.pojo.general.Cal_Servidor" lazy="false">
    <column name="codigo_servidor"/>
```

```
    </many-to-one>

   </class>
</hibernate-mapping>
```

1.2.22 Cal_controladorWeb.java (Paquete sgpi.Web)

```
package sgpi.web;

import java.io.FileNotFoundException;
import java.io.IOException;
import java.sql.Date;
import java.sql.ResultSet;
import java.sql.Timestamp;
import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.Properties;
import java.util.Set;
import java.util.StringTokenizer;

import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import net.sf.jasperreports.engine.JRDataSource;
import net.sf.jasperreports.engine.JRResultSetDataSource;
import net.sf.jasperreports.engine.JasperCompileManager;

import org.springframework.beans.factory.InitializingBean;
import org.springframework.web.bind.ServletRequestBindingException;
import org.springframework.web.bind.ServletRequestDataBinder;
import org.springframework.web.bind.ServletRequestUtils;
import
org.springframework.web.multipart.support.ByteArrayMultipartFileEditor;
import org.springframework.web.servlet.ModelAndView;
import
org.springframework.web.servlet.mvc.multiaction.MultiActionController;
import
org.springframework.web.servlet.view.jasperreports.JasperReportsPdfView;

import sgpi.AccesoBaseDatos3;
```

```
import sgpi.AppException;
import sgpi.db.AccesoDAO3;
import sgpi.estadisticas.Resultado;
import sgpi.pojo.general.Cal_Alarma;
import sgpi.pojo.general.CreaPdf;
import sgpi.pojo.general.Grabalarma;
import sgpi.pojo.general.Cal_Historial;
import sgpi.pojo.general.Cal_Mensaje;
import sgpi.pojo.general.Cal_Participante;
import sgpi.pojo.general.Cal_Proyecto;
import sgpi.pojo.general.Registro_Alarma;
import sgpi.pojo.general.Cal_Responsable;
import sgpi.pojo.general.Sendmail3;
import sgpi.pojo.general.Cal_Servidor;
import sgpi.pojo.general.Cal_TipoEnvio;
import sgpi.pojo.general.Cal_Copiar;
import com.lowagie.text.DocumentException;



public class Cal_ControladorWeb extends MultiActionController implements
            InitializingBean {

      AccesoBaseDatos3 accesoDB;

      protected void initBinder(HttpServletRequest request,
      ServletRequestDataBinder binder) throws Exception {
      binder.registerCustomEditor(byte[].class, new
      ByteArrayMultipartFileEditor());
      }

      public ModelAndView alarmaEditor(HttpServletRequest request,
                  HttpServletResponse response) throws
                  ServletException {
            String bandera = request.getParameter("accion");

            if ("guardar".equals(bandera)) {
                  Cal_Alarma alarma = new Cal_Alarma();
                  Cal_Mensaje mensaje = new Cal_Mensaje();
                  Cal_Responsable responsable = new
                  Cal_Responsable();
                  Cal_Servidor servidor = new Cal_Servidor();
                  Cal_TipoEnvio tipoenvio = new Cal_TipoEnvio();

                  Integer codigo_alarma =
                  ServletRequestUtils.getIntParameter(request,
                  "codigo_alarma", 0);
```

```
Integer codigo_mensaje =
ServletRequestUtils.getIntParameter(request,
"codigo_mensaje", 0);
Integer codigo_respon_alarma =
ServletRequestUtils.getIntParameter(request,
"codigo_respon_alarma", 0);
Integer codigo_servidor =
ServletRequestUtils.getIntParameter(request,
"codigo_servidor", 0);
ArchivoAdjunto objadjunto = new ArchivoAdjunto();
try {
        bind(request, objadjunto); //Carga en el
        objadjunto el archivo
} catch (Exception e) { }

if (codigo_mensaje != 0) {
        mensaje =
        getAccesoDB().getMensaje(codigo_mensaje);

}

        String xx =
        ServletRequestUtils.getStringParameter(request
        , "archivo_adjunto");

StringTokenizer t = new StringTokenizer(xx,"\\");
 int cuenta = t.countTokens();
 String rutas = "";
 String archiv = "";
 int ctdor =1;
while(t.hasMoreTokens()){
 if (ctdor<cuenta){
  rutas += t.nextToken()+"\\";
  }else
  archiv += t.nextToken();
          ctdor += 1;
  }
mensaje.setDescripcion_mensaje(ServletRequestUtils.getStrin
gParameter(request, "descripcion_mensaje"));
mensaje.setDescripcion_asunto(ServletRequestUtils.getStringParame
ter(request, "descripcion_asunto"));

mensaje.setArchivo_adjunto(objadjunto.getArchivo_adjunto());
mensaje.setNombre_archivo(archiv);
mensaje.setRuta_archivo(rutas);
getAccesoDB().guardaMensaje(mensaje);
if (rutas != "" && archiv != ""){
```

```
sgpi.pojo.general.Cal_Copiar.main(rutas,archiv);

}
        if (codigo_servidor != 0) {
        servidor = getAccesoDB().getServidor(codigo_servidor);
        }
        servidor.setDireccion_ip("127.0.0.1");
        getAccesoDB().guardaServidor(servidor);


        if (codigo_alarma != 0) {
        alarma = getAccesoDB().getAlarma(codigo_alarma);
        }
        Integer codigo_tipo_envio =
        ServletRequestUtils.getIntParameter(request,
        "codigo_tipo_envio");
                tipoenvio =
                getAccesoDB().getTipoEnvio(codigo_tipo_envio);
                alarma.setObjtipoenvio(tipoenvio);

                DateFormat defaultDate =
                DateFormat.getDateInstance(3);
                Calendar cal2 = Calendar.getInstance();
                cal2.setTime(new java.util.Date());  // Fecha Actual para
                comparacion tipo Date
                java.util.Date fwindows = cal2.getTime();
                String fecha_inicio_act =defaultDate.format(fwindows);
                boolean er = false;
                Timestamp fechai= null;
                SimpleDateFormat d1 = new
                SimpleDateFormat("dd/MM/yyyy");
                try {
                        String texto = fecha_inicio_act;
                        fechai = new
                        Timestamp(d1.parse(texto).getTime());

                } catch (ParseException e) {
                        er = true;
                }
                        if (er)System.out.println ("Problemas Con
                        Registro de Fechas ");else{
                        alarma.setFecha_inicio_act(fechai);
                        alarma.setFecha_fin_act(formatDate(request,
                        "fecha_fin_act"));
alarma.setRepeticiones(ServletRequestUtils.getStringParameter(requ
est, "repeticiones"));
alarma.setObjmensaje(mensaje);
alarma.setObjservidor(servidor);
```

```
alarma.setEstado("P");
getAccesoDB().guardaAlarma(alarma);
}
        if (codigo_respon_alarma != 0) {
        responsable =
        getAccesoDB().getResponsable(codigo_respon_alarma); }
        Integer codigo_participante =
        ServletRequestUtils.getIntParameter(request,
        "codigo_participante");
        Cal_Participante objparticipante
        =getAccesoDB().getParticipante(codigo_participante);
        responsable.setObjpersona(objparticipante.getObjpersona());
        Integer codigo_proyecto =
        ServletRequestUtils.getIntParameter(request,
        "codigo_proyecto");
        Cal_Proyecto objproyecto=
        getAccesoDB().getProyecto(codigo_proyecto);
        responsable.setObjproyecto(objproyecto);
        responsable.setObjalarma(alarma);
        Integer codigo_origen =
        ServletRequestUtils.getIntParameter(request,
        "codigo_origen");
        responsable.setCodigo_origen(codigo_origen);
        getAccesoDB().guardaResponsable(responsable);
        }

        else {
                Integer codigo_proyecto =
                ServletRequestUtils.getIntParameter(
                request, "codigo_proyecto", 0);
                List lista_proyectos =
                getAccesoDB().consultaProyecto();
                Set lista_participantes =
                getAccesoDB().consultaParticipante(
                codigo_proyecto);
                List lista_personal = getAccesoDB().getPersonas();
                List lista_tipoenvio =
                getAccesoDB().consultaTipoEnvio();
                System.out.println("Efecto Carga de Datos a los
                Combos"
                + lista_proyectos.size());
                Cal_Proyecto p = new Cal_Proyecto();
                HashMap map = new HashMap();
                map.put("lista_proyectos", lista_proyectos);
                map.put("lista_participantes", lista_participantes);
                map.put("lista_tipoenvio", lista_tipoenvio);
                map.put("lista_personal", lista_personal);
                return new ModelAndView("alarmaForm", map);
```

```
}

Integer codigo_proyecto =
ServletRequestUtils.getIntParameter(request,
"codigo_proyecto", 0);
List lista_proyectos = getAccesoDB().consultaProyecto();
Set lista_participantes = getAccesoDB().consultaParticipante(
codigo_proyecto);
List lista_personal = getAccesoDB().getPersonas();
List lista_tipoenvio = getAccesoDB().consultaTipoEnvio();
System.out.println("Efecto Carga de Datos a los Combos"
+ lista_proyectos.size());
Cal_Proyecto p = new Cal_Proyecto();
HashMap map = new HashMap();
map.put("lista_proyectos", lista_proyectos);
map.put("lista_participantes", lista_participantes);
map.put("lista_tipoenvio", lista_tipoenvio);
map.put("lista_personal", lista_personal);
return new ModelAndView("alarmaForm", map);
}
private Date formatDate(HttpServletRequest request, String
parameter) {
Date fecha = null;
SimpleDateFormat d = new SimpleDateFormat("dd/MM/yyyy");

try {
        String texto =
        ServletRequestUtils.getStringParameter(request,
        parameter);
        fecha = new Date(d.parse(texto).getTime());
} catch (ParseException e) {
} catch (ServletRequestBindingException e) {
                        }
return fecha;
}

private Timestamp formatTimestamp(HttpServletRequest
request, String parameter, String parametro2) {
SimpleDateFormat d;
String texto, hora;
Timestamp fecha = null;

try {
        texto =
        ServletRequestUtils.getStringParameter(request,param
        eter);
```

```
                hora =
        ServletRequestUtils.getStringParameter(request,parametro2);
                if (hora != null && hora.length() > 0) {
                texto = texto + " " + hora;
                d = new SimpleDateFormat("dd/MM/yyyy h:mm");
                                }
                else {
                        d = new SimpleDateFormat("dd/MM/yyyy");
                //Solo fecha
                }

                fecha = new Timestamp(d.parse(texto).getTime());

        } catch (ParseException e) {
        } catch (ServletRequestBindingException e) {

        }
        return fecha;
}

// deseo presentar la hora
private Date formatHora(HttpServletRequest request, String
parameter) {
        Date hora = null;
        SimpleDateFormat d = new SimpleDateFormat("h:mm a");
        try {
                String texto =
                ServletRequestUtils.getStringParameter(request,
                                parameter);
                hora = new Date(1, 1, 2000);// new
                Date(d.parse(texto).getTime());

        } catch (ServletRequestBindingException e) {
                // e.printStackTrace();
        }
        return hora;
}

public ModelAndView salirHandler(HttpServletRequest request,
                HttpServletResponse response) throws
ServletException, AppException {

        // Salir de la sesion de usuario
        request.getSession().invalidate();

        return indexHandler(request, response);
}
```

```
	public ModelAndView pruebafuncion(HttpServletRequest request,
			HttpServletResponse response) throws
			ServletException, AppException {
			request.getSession().invalidate();

		byte[] a = "datos del archivo adjunto
		peroksdlgjlkgsdjfgklsjdfgklsjdflskdf sin nombre".getBytes();

	return indexHandler(request, response);
	}





	/**
	 * Menú de la aplicación
	 */
	public ModelAndView indexHandler(HttpServletRequest request,
			HttpServletResponse response) throws
ServletException, AppException {


		return new ModelAndView("indexView");
	}

	public ModelAndView estadisticas(HttpServletRequest request,
			HttpServletResponse response) throws
ServletException, AppException {

		Integer codigo_proyecto =
ServletRequestUtils.getIntParameter(request, "codigo_proyecto", 0);
		List listar_proyectos = getAccesoDB().consultaProyecto();

	Map refData = new HashMap();
	refData.put("listar_proyectos", listar_proyectos);
    Resultado objResl = (Resultado)
getApplicationContext().getBean("objResultado");
    objResl.setCodigo_proyecto(codigo_proyecto);

    refData.put("objResultado", objResl);
		return new ModelAndView("estadistica", refData);
	}



	public ModelAndView alarmaHandler(HttpServletRequest request,
```

```
                        HttpServletResponse response) throws
ServletException, AppException {

                return new ModelAndView("alarmaForm");
        }

        public ModelAndView presentaAlarmas(HttpServletRequest request,
HttpServletResponse response)
        throws ServletException, AppException {

                Integer codigo_proyecto =
ServletRequestUtils.getIntParameter(request, "codigo_proyecto", 0);
                List result2 =
        getAccesoDB().presentaResponsables(codigo_proyecto);


                List lista_proyectos = getAccesoDB().consultaProyecto();
                Set lista_participantes = getAccesoDB().consultaParticipante(
                                codigo_proyecto);

                List lista_tipoenvio = getAccesoDB().consultaTipoEnvio();

                Cal_Alarma objalarma = getAccesoDB().getAlarma(49);
      Cal_Proyecto p = new Cal_Proyecto();
                HashMap map = new HashMap();
                map.put("lista_proyectos", lista_proyectos);
                map.put("lista_participantes", lista_participantes);
                map.put("lista_tipoenvio", lista_tipoenvio);
                map.put("result2", result2);
                //envia a seccion de impresion
                String banderaimpr = request.getParameter("imprimir");
                if ("imprimir".equals(banderaimpr)) {
                                return reporteriaHandler(request, response);

                }
      //prueba grabacion de alarma
                byte[] a = "datos del archivo adjunto
peroksdlgjlkgsdjfgklsjdfgklsjdflskdf sin nombre".getBytes();
                //getAccesoDB().regAlarma("hola","como estan
hoy",2,2,"12/12/2007","22/12/2007","1",2,2,"d:\\","sendmail2.java");



     return new ModelAndView("presenta", map);

        }

         /**
```

```
     * Reporteria de la aplicación
     */
    public ModelAndView reporteriaHandler(HttpServletRequest request,
HttpServletResponse response)
        throws ServletException, AppException {


            ServletContext context = request.getSession().getServletContext();
            Map model = new HashMap();
            model.put("FECHA_HOYDIA", (new java.util.Date()).toLocaleString());
            model.put("BASE_PATH", context.getRealPath("/images/"));

            String sql = "select
codigo_mensaje,descripcion_mensaje,descripcion_asunto from
cal_mensaje";

            ResultSet result = getAccesoDB().executeQuery(sql);
            JRDataSource dataSource = new JRResultSetDataSource(result);
            model.put("dataSource", dataSource);

            try {
                        JasperCompileManager.compileReportToFile(

            context.getRealPath("/reporte/detalle.jrxml"),

            context.getRealPath("/reporte/detalle.jasper"));
                  } catch (Exception e) {
                        e.printStackTrace();
                  }

                  //Headers de control
                  Properties ps = new Properties();
                  ps.put("Expires","0");
                  ps.put("Cache-Control", "must-revalidate, post-check=0, pre-
check=0");
                  ps.put("Pragma", "public");
                  ps.put("Content-Disposition", "attachment; filename=\"" +
"reporte.pdf" +"\"");

                  //Parametrizando varios Headers de control HTTP/1.0
                  JasperReportsPdfView pdfView = new
JasperReportsPdfView();
                  pdfView.setUrl("/reportes/detalle.jasper");
                  pdfView.setApplicationContext(getApplicationContext());
                  pdfView.setHeaders( ps );
                  return new ModelAndView(pdfView, model);

);
```

```
        }

        public void afterPropertiesSet() throws Exception {
                // TODO Auto-generated method stub
        }

        public AccesoBaseDatos3 getAccesoDB() {
                return accesoDB;
        }

        public void setAccesoDB(AccesoBaseDatos3 accesoDB) {
                this.accesoDB = accesoDB;
        }

}
```

CAPITULO 2

2. MANUAL DE USUARIO

Nuestro Modulo de Comunicación y alarmas consta de 3 pantallas las cuales seran descritas a continuación.

2.1 Pantalla de Inicio.

En esta pantalla usted encontrara un menú donde podra seleccionar las siguientes opciones (Ver figura 47):

- **Registrar Alarmas**.- Al dar clic en esta opción se visualizara una pantalla donde podra definir un mensaje (Ver figura 48).
- **Reporte de alarmas**.- Al dar clic en esta opción se visualizará una pantalla donde podra revisar en modo de reporte los mensajes definidos según el proyecto previamene seleccionado (ver figura 49).
- **Estadísticas**.- Al dar clic en esta opción se visualizara una pantalla donde podra generar en modo de reporte gráfico los mensajes definidos según le proyecto seleccionado (Ver figura 50).
- **Salir**.- Al dar clic en esta opción usted saldra de cuialquiera de la opciones antes ingresada y volvera al menú principal.



Figura 47

Comunicación y Alarmas

Registrar Alarmas | Reporte de Alarmas | Estadisticas | Salir

Proyectos: [11] -REPORTE DE FACTURAS

Para:

De: AMADA JUDITH GUERRERO AGUIRRE

Tipo de Envío: EMAIL Forma de Envío: Ninguno

Fecha de Envío:

Asunto:

Archivo Adjunto: Examinar...

Figura 48

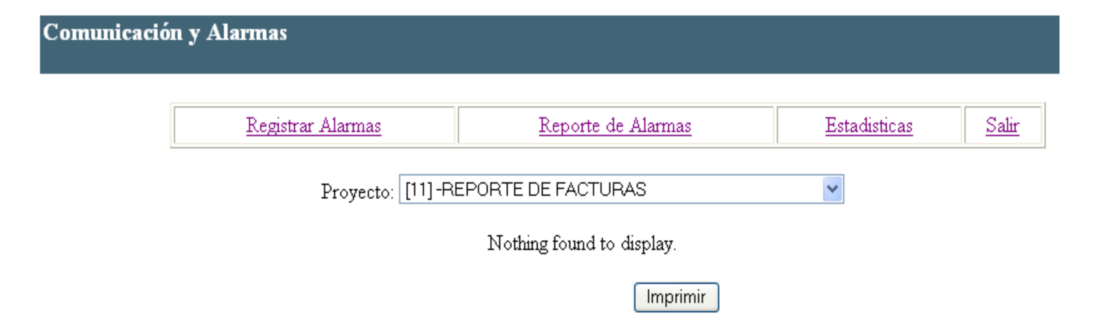


Figura 49

Comunicación y Alarmas

Registrar Alarmas	Reporte de Alarmas	Estadisticas	Salir

Codigo del Proyecto: [11] -REPORTE DE FACTURAS

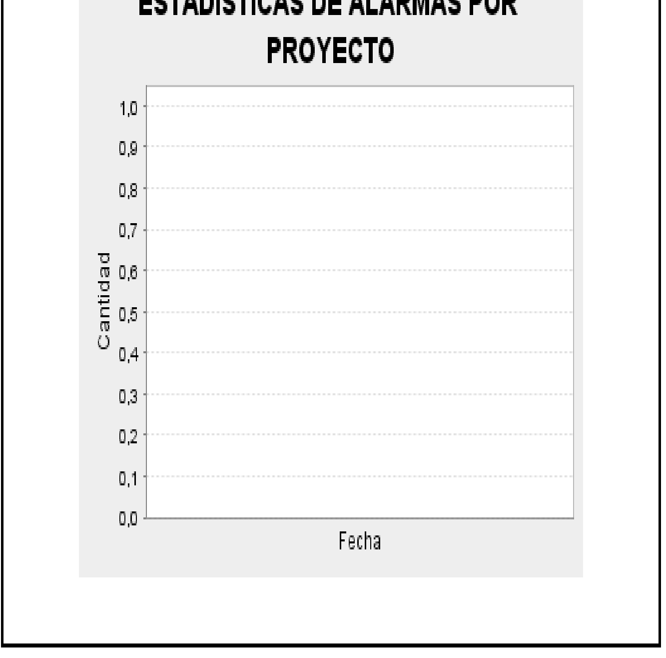


Figura 50

2.2 Pantalla Registrar Alarmas.

En esta pantalla definiremos los mensajes que se deseen enviar, la forma de hacerlo es la siguiente:

- Elegimos el proyecto que deseemos cargar de la lista que tenemos en la pantalla principal (ver figura 51).

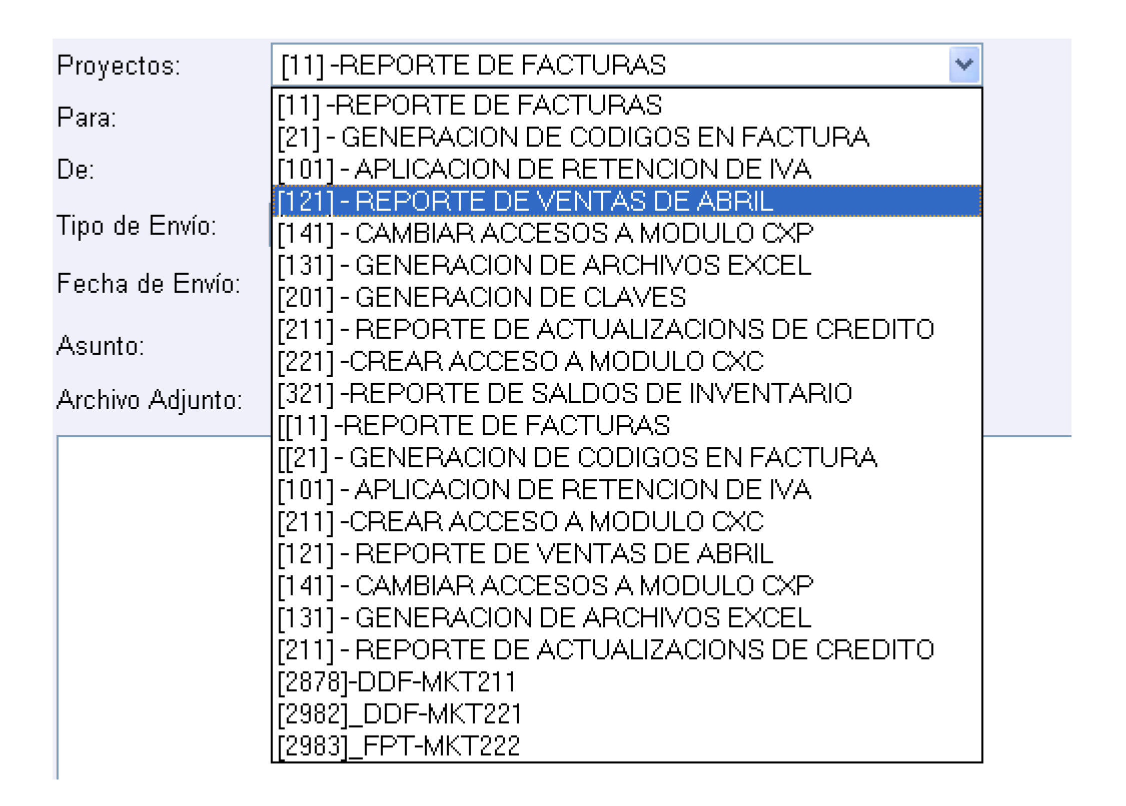


Figura 51

- Luego de elegir el proyecto se actualizara la pantalla automaticamente y se visualizara las personas que forman parte de ese proyecto (Ver figura 52).

El “Para”, es donde estaran las personas involucradas en ese proyecto y son a ellas a quienes podemos enviarles un mensaje.

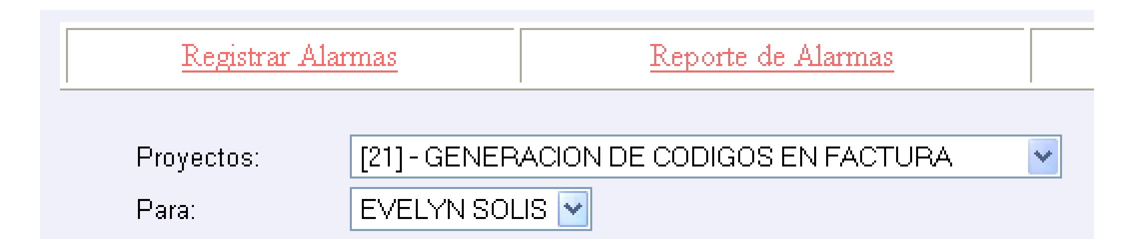


Figura 52

- El “De” es donde se define quien envía el mensaje (ver figura 53).

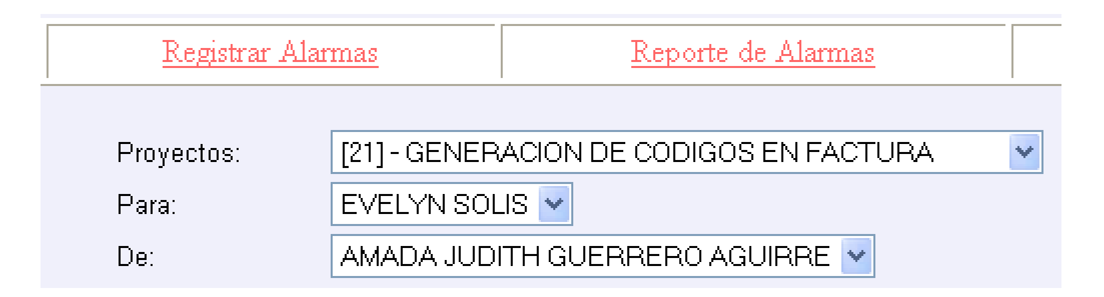


Figura 53

- El tipo de Envío es el como se enviará el mensaje ya sea por e-mail o sms y la forma de envío son la frecuencia de envío del mensaje ya sea diario o semanal (Ver figura 54 y 55).

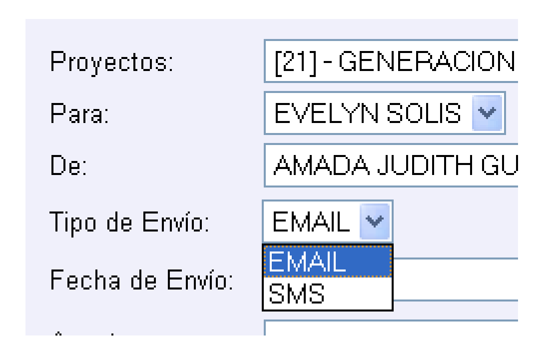


Figura 54

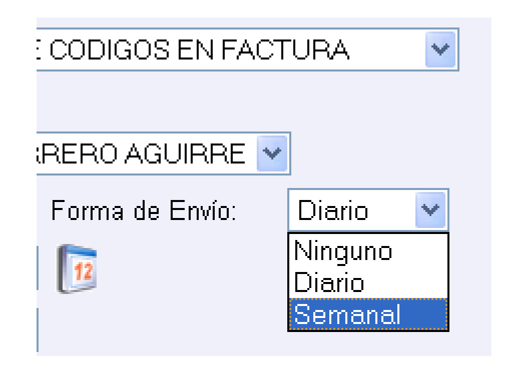


Figura 55

- En la opción de Fecha es donde definimos la fecha de cuando queremos que se envíe el mensaje(Ver figura 56).

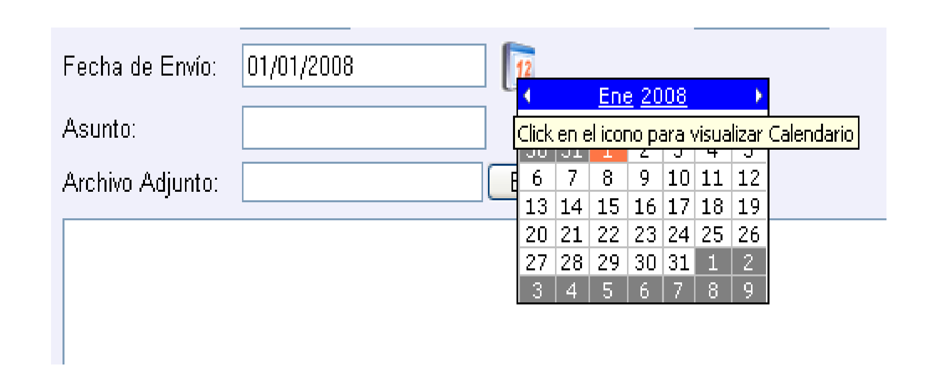


Figura 56

- En la opción de Asunto ponemos el tema referente al mensaje que deseamos enviar (ver figura 57).

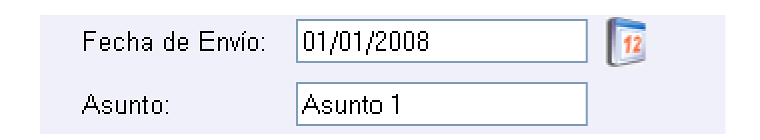


Figura 57

- En la opcion de Archivo Adjunto usted podra adjuntar algun tipo archivo que desee definir junto al mensaje.

Al dar clic en el boton examinar se visualizara una pantalla donde podra buscar el archivo a adjuntar, luego de seleccionarlo dar clic en abrir y se adjuntara el archivo (Ver figura 58 y 59).

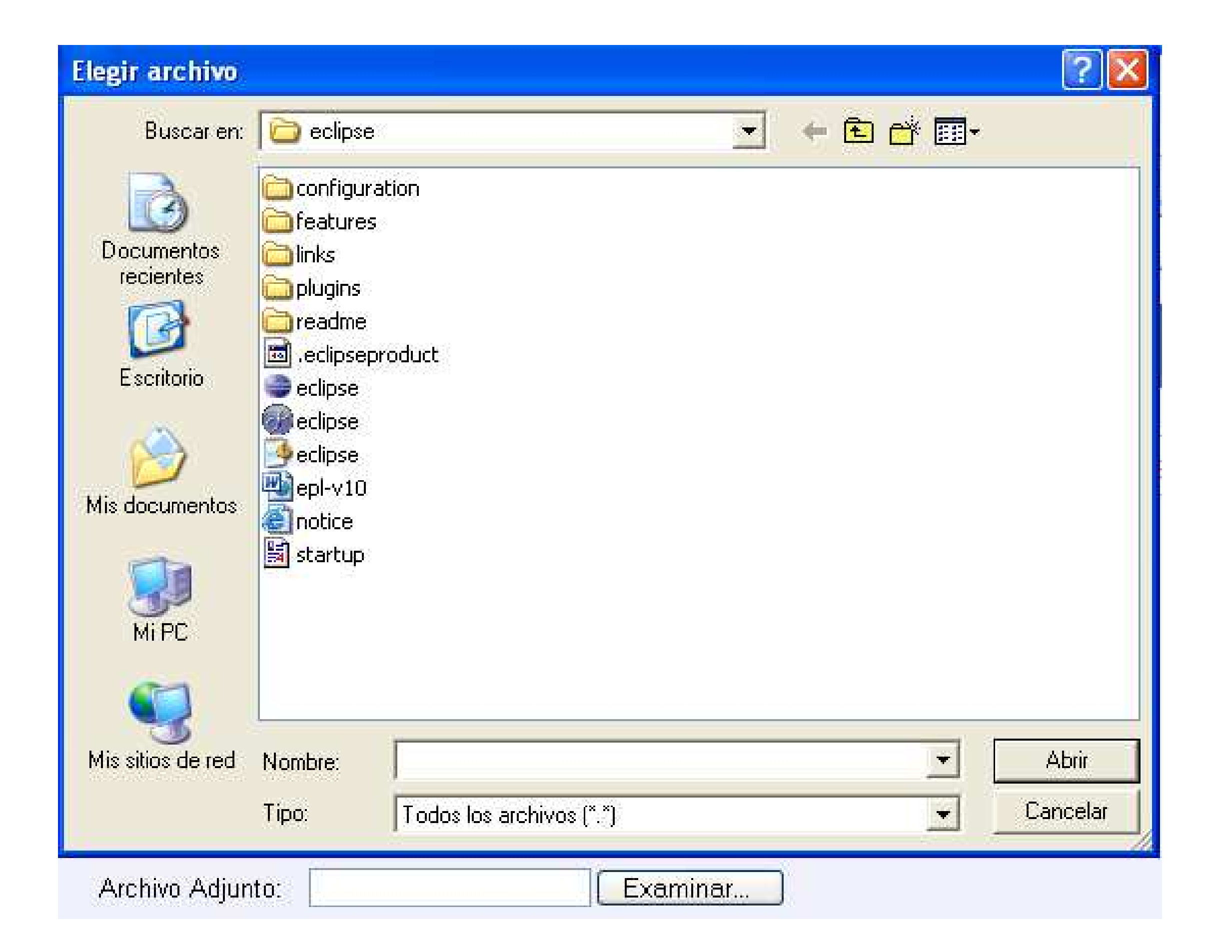


Figura 58

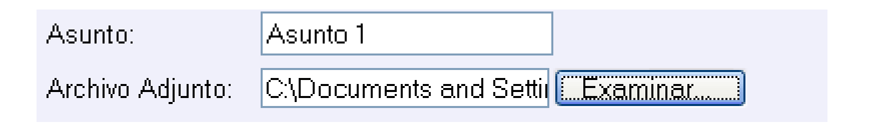


Figura 59

- En la opcion donde visualizamos el area mas grande de la pantalla, es donde definiremos el mensaje que deseemos enviar (Ver figura 60).



Figura 60

- Una vez que hemos definido todo lo necesario para el envío del mensaje damos clic en el boton de Guardar y le saldra un mensaje indicando que los datos que defiinió estan correctamente ingresados, luego da clic en ese mensaje y listo su mensaje ha sido enviado (ver figura 61).

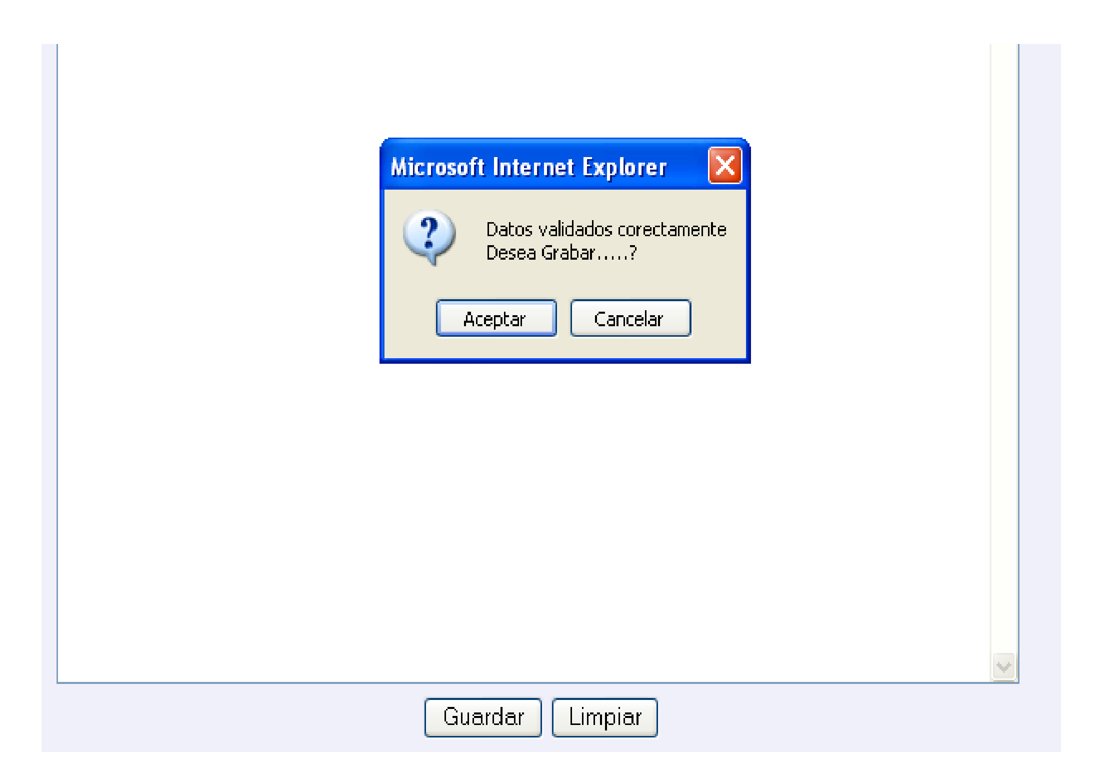


Figura 61

2.3 Pantalla de Reporte de Alarmas.

En esat pantalla visualizamos los mensajes definidos por proyecto, la forma de hacerlo es la siguiente:

- Primero debemos elegir el proyecto que deseemos visualizar (ver figura 62).

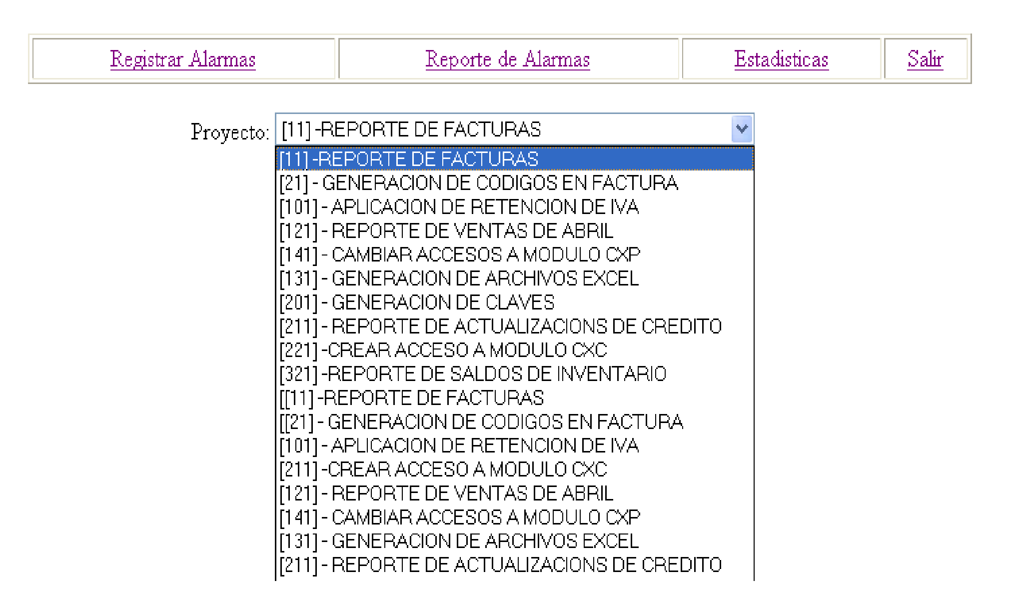


Figura 62

- Luego se cargaran los mensaje que hemos definido para ese proyecto (Figura 63), si elegimos algun proyecto que no tiene definido ningun mensaje, entonces no se visualizara nada en la pantalla.

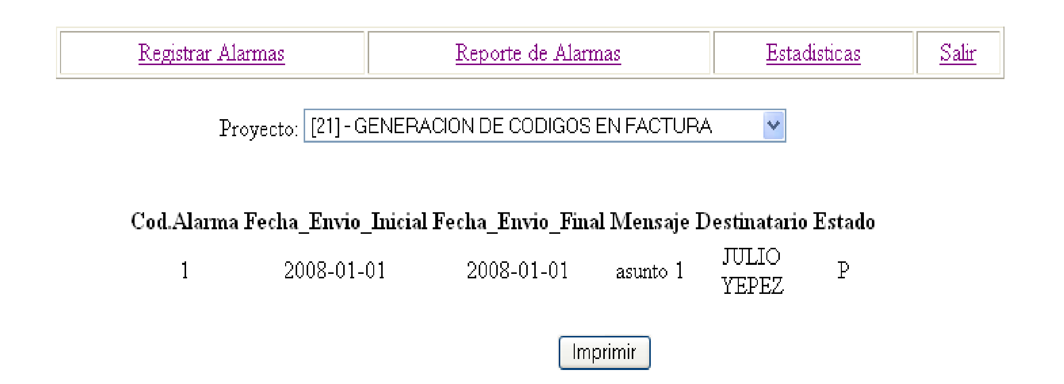


Figura 63

2.4 Pantalla de Estadisticas

- En esta pantalla visualizaremos en un cuadro estadístico cuantos mensajes han sido enviados en un determinado proyecto, de igual manera que en la anterior pantalla, primero debemos elegir el proyecto del que deseamos visualizar la estadística, luego de eso cargara la información, si no existen datos definidos, no se podra visualizar nada (Ver figura 64).



Figura 64

CONCLUSIONES

El Modulo de Comunicación y Alarmas es la vía de comunicación que existirá en el Sistema de Gestión de Proyectos Informáticos, y mantendrá:

- Una fluida comunicación entre todas las personas que forman parte de los diferentes proyectos.
- Medidas correctivas a tiempo, ya que al informar de la evolución del proyecto a los respectivos responsables ayudará a tomar la mejor decisión para superar el problema.
- Permitirá informar al personal que ha sido asignado ha un proyecto.
- Permitirá informar al Líder del proyecto sobre le estado en que se encuentra el proyecto, cambios de Hitos o de cronograma.
- Permitirá informar al personal sobre resultados de evaluaciones realizadas en la empresa.
- Permitirá Informar al Líder del proyecto sobre la aprobación y puesta en marcha de determinado proyecto.

RECOMENDACIONES

Técnicas

El usuario a quien se le envíe una alarma deberá tener una cuenta de correo en su computador personal, previo a esto se debe definir un usuario por el cual fluirán los mensajes a su destino, este usuario debe ser creado en el servidor de correos el cual necesitará de un computador con especificaciones normales.

Generales

Para un buen funcionamiento del módulo de Comunicación y Alarmas es necesario tener ingresado correctamente la cuenta de e-mail y el número celular en la tabla donde se definen a las personas que forman parte de la empresa ya que esta debe ser igual a la definida en el servidor de correos.

Bibliografía:

Libro de Ingeniería de Software Autor: Roger Pressman.

http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=pmd

http://www.adslayuda.com/servidores-servidor_smtp.html

http://geneura.ugr.es/~gustavo/css/css.html

Http://Java.sun.com/products/jms/tutorial/index.html

http://ww.programacion.com/java/tutorial/j2ee/

http://es.wikipedia.org/wiki/Java_EE

http://www.eclipse.org/downloads/

www.lawebdelprogramador.com

http://www.myeclipseide.com

http://www.ajpdsoft.com

www.monografias.com

http://www.autentia.com